Modèles linéaires pour données fonctionnelles multivariées

Issam-Ali Moindjié

Équipe projet MODAL, Centre INRIA de l'Université de Lille
Laboratoire Paul Painlevé (UMR 8524), Université de Lille
École doctorale MADIS, Université de Lille

Thèse présentée par **Issam Ali MOINDJIÉ**
Soutenue le **18 décembre 2023**

En vue de l'obtention du grade de docteur de l'Université de Lille
Discipline **Mathématiques et leurs interactions**
Spécialité **Statistique**

Modèles linéaires pour données fonctionnelles multivariées

Thèse dirigée par : Cristian PREDA — directeur
Sophie DABO-NIANG — co-directrice

Membres du jury

Rapporteurs

Mustapha RACHDI	Professeur à l'Université Grenoble Alpes
Yousri SLAOUI	Maître de conférences (HDR) à l'Université de Poitiers

Examinateurs

Présidente du jury	Ndeye NIANG-KEITA	Professeure au Conservatoire National des Arts et Métiers
	Azzouz DERMOUNE	Professeur à l'Université de Lille

Encadrants

Cristian PREDA	Professeur à l'Université de Lille
Sophie DABO-NIANG	Professeure à l'Université de Lille

Résumé

Le cadre méthodologique de cette thèse est l'analyse de données fonctionnelles. Nous nous intéressons particulièrement au problème de la prédiction d'une variable réelle ou catégorielle à l'aide de variables fonctionnelles multivariées.

Dans la littérature existante, les méthodes proposées ont souvent recours au cadre restrictif du domaine unique. Il signifie que chaque dimension de la variable fonctionnelle multivariée a le même domaine de définition. Cette hypothèse limite leurs utilisations pour un certain nombre de domaines d'application.

En effet, l'émergence des nouvelles technologies de collecte et de stockage de données a permis l'observation de plusieurs caractéristiques fonctionnelles, parfois de type différent, pour un même individu statistique.

Pour répondre à la problématique de prédiction avec ce type de variables, nous proposons des méthodes basées sur la régression PLS : MFPLS et TMFPLS. La première est une extension de l'algorithme PLS au cas des données fonctionnelles multivariées explicatives, où les dimensions sont potentiellement définies sur différents domaines. Cette méthode peut être utilisée pour la régression et la classification (supervisée) binaire. La deuxième méthode : TMFPLS, est un arbre de décision qui permet de répondre à des tâches de classification plus complexes (relation non-linéaire entre la variable à prédire et les variables explicatives, plusieurs classes tolérées).

Ces méthodes peuvent être utilisées dans divers domaines d'applications, cependant, les interpréter devient difficile lorsque les données explicatives ont de nombreuses dimensions. C'est le cas typiquement lorsque plusieurs capteurs sont utilisés pour mesurer une variable fonctionnelle suivant plusieurs localisations. Ou plus généralement, lorsque l'on a à faire à des données fonctionnelles répétées.

Dans ce cas, nous présentons des méthodes parcimonieuses basées sur la pénalité fusion permettant d'obtenir une meilleure interprétation des modèles. Les applications sur des données simulées et données réelles (EEG, ECG, etc.) ont permis de démontrer la bonne performance de nos méthodes.

Abstract

In this thesis, we are interested in the problem of predicting a real or categorical variable using multivariate functional variables.

In the existing literature, the proposed methods often assume the case of a single domain. This means that each dimension of the multivariate functional variable has the same domain of definition. This assumption restricts their use to a limited number of applications.

Indeed, technological advances in data collection and storage have made it possible to observe several functional characteristics, sometimes of different natures, for the same statistical individual.

To solve the prediction problem with this type of variables, we proposed two methods inspired by the PLS regression : MFPLS and TMFPLS. The first one is an extension of the PLS algorithm to the case of explanatory multivariate functional data, where the dimensions are potentially defined on different domains. This method can be used for regression and binary classification. The second method : TMFPLS, is a decision tree which can be used for more complex classification tasks (non-linear relationship between the target variable and the predictors, multiclass classification).

These methods can be used for a wide range of applications ; however, interpreting their results becomes difficult when the predictors have many dimensions. This is typically the case when many sensors are used to measure a functional variable in several locations. Or more generally, when it comes to repeated functional data.

In this case, we present parsimonious methods based on the fusion penalty, to obtain more interpretable models. Applications on simulated data and real data (EEG, ECG, etc.) have demonstrated the good performance of our methods.

Remerciements

Tout d'abord, je voudrais remercier mes directeurs de thèses : Cristian Preda et Sophie Dabo, pour les échanges, les corrections et les conseils qui m'ont servis durant ces trois dernières années et qui me serviront encore dans le futur.

Merci aussi aux membres du jury pour leurs présences et leurs remarques pertinentes qui m'ont permis d'améliorer ce travail. Je les remercie également pour nos discussions enrichissantes ; elles m'ouvrent de nouveaux horizons d'investigations.

Je remercie la Région Hauts-de-France d'avoir co-financé mon contrat doctoral. Je remercie également l'équipe GRAMFC du CHU Amiens Picardie pour leur hospitalité lors de mes visites et leurs pédagogies pour m'expliquer les phénomènes à l'origine des données d'électroencéphalogramme.

Je tiens également à remercier mes collègues de Diagrams Technologie : Florent Dewez et Quentin Grimonprez, de leurs disponibilités et pour nos conversations scientifiquement enrichissantes.

Merci à tous mes proches : ceux qui ont partagé mes échecs et mes victoires. Je les remercie pour leurs encouragements, leurs conseils et d'avoir été si compréhensifs durant mes absences, et ce même quand j'étais avec eux. Merci à Ismaël, Honoré, Faiyat, Emmanuel, Rayan, Amir, Paul, Ramzi, Bilal, Farah, Inzlat, Massim, Naima, Mouayad, Cheikh, Doumbaly (et ceux que j'oublie).

Il y a aussi mes collègues qui, peut-être parce que nous vivions la même chose, ont eu les bons mots. Les pauses cafés du "Syndicat" et les sorties après le boulot vont me manquer. Merci à Etienne, Wilfried, Eglantine, Filippo, Ernesto, Myriam, Yaroslav, Clarisse, Louise, Axel, François, Camille, Rim, Paguidame, Wilhem et Rachid.

Bien sûr, merci à mes parents pour leurs sacrifices consentis et leurs conseils qui m'ont guidés depuis mon départ au Sénégal jusqu'à l'obtention de ce diplôme. Tout ça n'aurait pas été possible sans vous, sans votre éducation, sans votre discipline, sans votre amour.

Enfin, je te remercie Marie pour… La liste serait trop longue à énumérer ici. Je me contenterai de te remercier d'avoir vécu cette aventure à mes côtés et d'être, depuis quelques années maintenant, ma lumière, mon espoir et l'autre voix de ma conscience.

Table des matières

Table des figures

Liste des tableaux

Introduction

En statistique, il est courant de travailler avec des variables aléatoires réelles. Celles-ci peuvent être de deux types principaux : continues, prenant des valeurs dans une plage de nombres réels (par exemple, le poids d'un individu), ou discrètes, prenant des valeurs dans un ensemble dénombrable.

Cependant, le développement des technologies de collecte de données a considérablement élargi le paysage des données, en particulier avec l'avènement des données massives (big data). De nos jours, il est possible d'obtenir des données de manière quasi continue grâce à des capteurs, des dispositifs en ligne, etc. Certaines de ces données ne correspondent pas au modèle classique de variables aléatoires réelles, mais plutôt à des variables aléatoires fonctionnelles, où les observations sont des fonctions continues au lieu de valeurs ponctuelles. Les observations de variables fonctionnelles sont communément appelées données fonctionnelles.

L'analyse de données fonctionnelles (Functional Data Analysis - FDA) est une discipline qui s'attache à développer des méthodes statistiques spécifiques pour traiter de telles données. En effet, les données fonctionnelles, généralement de grande dimension, présentent souvent des corrélations complexes, ce qui rend difficile l'application des méthodes statistiques classiques.

Cette thèse se situe dans le cadre de l'analyse de données fonctionnelles (FDA). Elle se concentre sur le problème de la prédiction d'une variable aléatoire réelle à l'aide de variables explicatives fonctionnelles. Ces dernières peuvent être regroupées dans un vecteur appelé variable fonctionnelle multivariée, où chaque composante est une variable fonctionnelle. Ce type de problème de prédiction se retrouve dans divers domaines, tels que la médecine, où des signaux d'électroencéphalogramme (EEG) peuvent être considérés comme des variables fonctionnelles multivariées, et où l'objectif est de prédire l'état d'un patient.

L'une des innovations de cette thèse est de proposer des méthodes qui permettent la prise en compte de différents types de variables fonctionnelles dans le modèle de prédiction. Par exemple, en plus des signaux EEG, il est possible d'inclure des images issues de l'imagerie par résonance magnétique (MRI, fMRI) comme variables explicatives.

Plus particulièrement, les modèles proposés dans cette thèse sont des modèles linéaires fonctionnels, connus pour leur interprétabilité et leur adaptabilité à une grande variété de situations.

Les contributions de la thèse se divisent en deux parties. La première se concentre sur le cas où les variables explicatives sont des variables fonctionnelles multivariées définies sur des domaines différents. Ce cadre permet de gérer différentes sources de données explicatives, cas de plus en plus courant avec les progrès technologiques. Par exemple, il permet de combiner des données d'EEG et de fMRI pour obtenir de meilleures prédictions.

La deuxième s'intéresse aux variables (fonctionnelles) explicatives répétées. Ces dernières sont des mesures répétées d'une même quantité physique selon différentes conditions. Lorsque ces mesures sont effectuées de manière continue, elles peuvent être considérées comme des variables fonctionnelles. La thèse propose des modèles parcimonieux pour

tenir compte de ces répétitions et faciliter l'interprétation des coefficients.

Plus en détail, la suite du manuscrit se structure de la façon suivante :

— Le chapitre 1 vise à établir les bases conceptuelles et les notions fondamentales nécessaires à une meilleure compréhension des contributions de la thèse. Il se focalise particulièrement sur les concepts clés de l'analyse de données fonctionnelles, tels que l'analyse en composante principale, la régression, la classification, etc. Ces concepts forment le socle sur lequel reposent les développements des contributions présentées dans la suite du document.

— La première partie des apports de la thèse s'ouvre par le chapitre 2. Il propose la première contribution : la régression PLS avec des données fonctionnelles multivariées. Il s'agit d'une extension de l'approche présentée dans Preda and Saporta (2002) pour le cas de données fonctionnelles multivariées définies sur des domaines différents. Parmi les résultats de notre travail, nous montrons que cette approche PLS peut être retrouvée en utilisant la régression PLS classique (Tenenhaus et al., 1995) et la régression PLS avec une variable fonctionnelle univariée (Preda and Saporta, 2002). Cette dernière relation est à l'origine d'une nouvelle procédure d'estimation. Les résultats des applications numériques montrent que notre approche est compétitive avec les méthodes existantes pour la régression et la classification binaire.

— À l'aide de l'analyse discriminante présentée dans le chapitre 2, le chapitre 3 introduit un arbre de décision qui prend pleinement en compte la structure fonctionnelle des données. Il est inspiré de l'arbre proposé dans Poterie et al. (2019) dans le cadre classique. Contrairement à l'analyse discriminante basée sur la régression PLS, il est adapté pour de la classification à plusieurs classes et il est beaucoup plus flexible, comme les applications numériques le prouvent. Ces dernières montrent en plus que le cadre multivarié fonctionnel permet d'intégrer le plan fréquentiel dans la discrimination. Le chapitre soulève aussi les problèmes liés à l'obtention des hyperparamètres de l'arbre, tout en proposant des méthodes pour y remédier.

— Le chapitre 4 rentre dans le deuxième axe des contributions de la thèse. Il traite du cas où les variables explicatives sont fonctionnelles et répétées. Cela se produit lorsque les dimensions de la variable explicative fonctionnelle représentent une même quantité mesurée suivant des conditions différentes. Nous avons considéré que ces conditions font partie d'un espace métrique, ce qui permet de définir une notion de distances entre elles. Ce cadre se prête naturellement au cas de figures où les conditions représentent des positions de capteurs. Les méthodologies présentées, inspirées des travaux de Land and Friedman (1997) et Tibshirani et al. (2005), font usage de la régression fonctionnelle avec la pénalité fusion pour estimer des modèles parcimonieux, faciles à interpréter. Les applications numériques montrent la compétitivité de ces méthodes.

— Le chapitre 5 vient clore le document. Il résume les contributions présentées et offre une discussion sur de potentielles perspectives.

Communications scientifiques

Articles

- Classification of multivariate functional data on different domains with Partial Least Squares approaches, en collaboration avec Sophie Dabo et Cristian Preda (décembre 2022)

  Accepté pour publication à *Statistics and Computing*
- Régressions linéaires fusions sur données spatiales fonctionnelles (juillet 2023)

  54es Journées de Statistique de la SFdS.
- Fusion regression methods with repeated functional data, en collaboration avec Cristian Preda et Sophie Dabo (décembre 2023).

  Soumis à CSDA et preprint disponible sur arxiv [1] et hal [2].

Communications orales

- Analyse de données fonctionnelles pour l'identification des biomarqueurs en EEG et en MEG chez les prématurés et les fœtus
  - Méthodes Aléatoires pour les Sciences de la Santé et de l'Environnement (MASSE'21)- IRN-AFRIMATH 14 -16 décembre 2021, Université Cheikh Anta Diop.
- Classification of multivariate functional data (defined on different domains) using PLS approach
  - 15th International Conference of the ERCIM WG on Computational and Methodological Statistics (CMStatistics 2022)- 17-19 décembre 2022- King's College London.
- Classification of multivariate functional data (defined on different domains)

  Séminaire de l'équipe Modal-14 février 2023- Centre Inria de l'Université de Lille.
- Régressions linéaires fusions sur données spatiales fonctionnelles

  54es Journées de Statistique - 3 au 7 juillet 2023- Université libre de Bruxelles.

1. https://arxiv.org/abs/2308.01747
2. https://hal.science/hal-04176783

Chapitre 1

Concepts fondamentaux

Ce chapitre pose le cadre méthodologique des contributions, celui de l'analyse de données fonctionnelles (FDA). Il présente les concepts fondamentaux et les travaux de la littérature essentiels à la compréhension du manuscrit. Le chapitre est organisé en trois parties. La première est consacrée à la définition globale des données fonctionnelles, en particulier celles qui sont multivariées et à leur expansion en bases de fonctions. La deuxième porte sur l'analyse en composantes principales fonctionnelles, technique centrale de la FDA. Enfin, le chapitre présente un état de l'art des méthodes d'apprentissage supervisées dans le cadre fonctionnel.

1.1 Données fonctionnelles

Les développements technologiques ont permis la collection de données sous forme de courbes, d'images ou de formes complexes, et ce, dans plusieurs domaines : santé, environnement, physique, biologie, etc. Ces avancées technologiques ont donné naissance à une branche de la statistique dénommée "Functional Data Analysis (FDA)".

Les premiers travaux sur la FDA remontent aux années 1950, bien que n'utilisant pas cette appellation (Rao (1958), Grenander (1950)). En effet, les termes "données fonctionnelles" et "variables fonctionnelles" sont apparus plus récemment, dans les années 1980-90 et ont été popularisés, entre autres, par les travaux suivants : Ramsay (1982), Ramsay and Dalzell (1991) et Ramsey and Silverman (2005). Depuis ces premiers travaux, la FDA a connu diverses contributions théoriques et appliquées. Une vision plus globale de celles-ci est donnée par les livres : Ferraty and Vieu (2006), Ramsey and Silverman (2005), Kokoszka and Reimherr (2017), et les articles : Wang et al. (2016), Koner and Staicu (2023).

La donnée fonctionnelle étant au cœur de la FDA, il convient de la définir pour mieux comprendre ce domaine de la statistique. Une donnée fonctionnelle est considérée comme une observation d'une variable aléatoire fonctionnelle X, définie sur un espace de probabilité $(\Omega, \mathcal{A}, \mathbb{P})$, et prenant ses valeurs dans un espace de fonctions (de dimension éventuellement infinie) $\mathcal{H}$. Un échantillon de données fonctionnelles désigne alors les n-réalisations $(x_1, \ldots, x_n)$ de X.

Plusieurs types d'espaces fonctionnels $\mathcal{H}$ ont été considérés dans la littérature. Par exemple, dans Bosq (2000), les auteurs s'intéressent aux données fonctionnelles à valeurs dans des espaces de Banach. Pour les espaces semi-métriques, on peut se référer à Ferraty and Vieu (2006). Dans Ramsey and Silverman (2005), les auteurs s'intéressent au cas où $\mathcal{H}$ est un espace de Hilbert. C'est dans ce cadre que s'inscrit notre étude, car c'est le plus largement étudié.

Dans ce dernier contexte, $\mathcal{H}$ est principalement défini comme :

a- $\mathcal{H} = L_2(\mathcal{T})$, où $L_2(\mathcal{T})$ est l'espace des fonctions de carré intégrable de $\mathcal{T}$ vers $\mathbb{R}$. Le domaine $\mathcal{T}$ est un segment compact de $\mathbb{R}^d$, où $d = 2$ pour une image et $d = 1$ pour un signal à une dimension.

b- $\mathcal{H} = L_2(\mathcal{T}_1) \times \ldots \times L_2(\mathcal{T}_p)$. De manière similaire, les domaines $\mathcal{T}_j$, $j = 1, \ldots, p$, sont des intervalles compacts de $\mathbb{R}^{d_j}$, avec $d_j \in \mathbb{N}^*$.

Le premier point **(a)** désigne le cas d'une variable fonctionnelle univariée. Depuis le travail de Ramsey and Silverman (2005), plusieurs contributions théoriques et appliquées

ont vu le jour pour ce type de variables. Une partie de ces contributions sera discutée dans la suite. Néanmoins, pour une vision détaillée, le lecteur peut se référer à l'article de synthèse Wang et al. (2016).

Dans le second point **(b)**, la variable X est dite fonctionnelle multivariée à p-dimensions. D'après Koner and Staicu (2023), ce cadre rentre dans la "seconde génération" de la FDA. Notre travail se concentre principalement sur ce dernier point, dont **(a)** est évidemment un cas particulier ($p = 1$).

Il est important de noter que la définition **(b)** de l'espace de fonctions $\mathcal{H}$, inspirée du travail de Happ and Greven (2018), n'est pas commune. Plus précisément, elle désigne le cas où les p-dimensions de X (notées $X^{(j)}$, $j = 1, \ldots, p$) sont observées sur des domaines $(\mathcal{T}_j,\ j = 1, \ldots, p)$ différents.
La plupart des travaux, lorsque X est multivariée, suppose que $X^{(j)}$, $j = 1, \ldots, p$ sont à valeurs dans le même espace de fonctions. Ce qui revient à considérer que $\mathcal{T}_1 = \ldots = \mathcal{T}_p = \mathcal{T}$. L'avantage de la définition choisie est que sa généralité permet de considérer des données fonctionnelles de plusieurs types (ex. images et séries temporelles), offrant une variété d'applications aux méthodes proposées.

L'espace hilbertien $\mathcal{H}$ admet comme produit scalaire $\langle , \rangle_{\mathcal{H}}$

$$\langle f, g \rangle_{\mathcal{H}} = \sum_{j=1}^{p} \int_{\mathcal{T}_j} f^{(j)}(t) g^{(j)}(t) dt$$

où $f = \left(f^{(1)}, \ldots, f^{(p)}\right)^\top$ et $g = \left(g^{(1)}, \ldots, g^{(p)}\right)^\top$ sont des fonctions dans $\mathcal{H}$. En particulier, pour X univariée ($p = 1$), on a que :

$$\langle f, g \rangle_{\mathcal{H}} = \int_{\mathcal{T}} f(t) g(t) dt$$

Notons la covariance et l'espérance de la variable aléatoire X respectivement par μ et Γ.

- L'espérance est une fonction dans $\mathcal{H}$,

  $$\mu = \begin{pmatrix} \mu^{(1)} & \ldots & \mu^{(p)} \end{pmatrix}^\top,$$

  où $\mu^{(j)}(t) = \mathbb{E}(X^{(j)}(t), t \in \mathcal{T}_j, j = 1, \ldots, p$.
- La covariance (ou opérateur de covariance) de X s'écrit comme une matrice de fonctions,

  $$\Gamma = \begin{pmatrix} \Gamma_{1,1} & \ldots & \Gamma_{1,p} \\ \vdots & \ldots & \vdots \\ \Gamma_{l,1} & \ldots & \Gamma_{l,p} \\ \vdots & \ldots & \vdots \\ \Gamma_{p,1} & \ldots & \Gamma_{p,p} \end{pmatrix} \tag{1.1}$$

  où

  $$\Gamma_{k,l}(s,t) = \text{Cov}\left(X^{(k)}(s), X^{(l)}(t)\right),\ s \in \mathcal{T}_k, t \in \mathcal{T}_l$$

  $k = 1, \ldots, p$ et $l = 1, \ldots, p$.
  En particulier, lorsque $p = 1$, elle est donnée par

  $$\Gamma(s,t) = \text{Cov}\left(X(s), X(t)\right),\ s \in \mathcal{T}, t \in \mathcal{T} \tag{1.2}$$

Pour plus de détails sur l'opérateur de covariance dans la configuration **b**, l'article Happ and Greven (2018) en fournit une étude approfondie.
Sans perte de généralités, nous supposons que la v.a.f (variable aléatoire fonctionnelle) X est de moyenne nulle, ce qui a pour but de simplifier l'écriture.

$$\mu^{(j)}(t) = 0, t \in \mathcal{T}_j$$

pour $j = 1, \ldots, p$.

En pratique, les réalisations d'une variable fonctionnelle sont généralement observées de manière discrète. En effet, les appareils de mesures ne peuvent pas enregistrer en continu. Les réalisations de X sont souvent observées en un nombre fini (souvent très grand) de points représentés par des grilles $\tilde{\mathcal{T}}_j$:

$$\tilde{\mathcal{T}}_j \subset \mathcal{T}_j$$

pour $j = 1, \ldots, p$.

Nous considérons que les données sont observées de manière dite régulière. Cela signifie que les grilles $\tilde{\mathcal{T}}_j$ sont les mêmes pour toutes les observations et leurs cardinaux notés k_j $j = 1, \ldots, p$ sont suffisamment grands pour reconstruire la forme fonctionnelle des observations.

Dans certaines situations, il est possible que k_j soit petit et que $\tilde{\mathcal{T}}_j$ dépende de l'observation i ($i = 1, \ldots, n$). Ce cas de figure est étudié, par exemple, dans Yao et al. (2005), Zhang and Wang (2016).

Les données sont observées sur des grilles discrètes plutôt que sur des intervalles compacts, ce qui contraste avec la définition d'une v.a.f. Ainsi, la reconstruction des observations discrètes en fonctions doit être réalisée au préalable de l'analyse fonctionnelle. Pour cela, la technique la plus utilisée est l'expansion dans une base de fonctions (Ramsey and Silverman, 2005).

Représentation dans une base de fonctions

L'expansion dans une base de fonctions suppose que les réalisations de X peuvent être résumées de façon satisfaisante à l'aide d'un système de fonctions appelé "base". Ce système n'est pas à proprement parler une base dans $\mathcal{H}$, mais un ensemble composé de fonctions linéairement indépendantes.

Lorsque la variable est multivariée, pour chaque dimension j ($j = 1, \ldots, p$), un système de fonctions $\psi_1^{(j)}, \ldots, \psi_{M_j}^{(j)}$ est considéré. Les fonctions $\psi_1^{(j)}, \ldots, \psi_{M_j}^{(j)}$ sont linéairement indépendantes dans $L_2(\mathcal{T}_j)$ pour $j = 1, \ldots, p$.

Pour reconstruire les fonctions associées aux réalisations x_i $i = 1, \ldots, n$, l'objectif de la méthode est d'estimer les scores (coefficients) (a) permettant d'écrire :

$$x_i^{(j)} \simeq \sum_{m=1}^{M_j} a_{i,m}^{(j)} \psi_m^{(j)}, \quad i = 1, \ldots, n \; j = 1, \ldots, p \tag{1.3}$$

où $a_{i,m}^{(j)} \in \mathbb{R}$ sont les scores de $x_i^{(j)}$ sur $\psi_m^{(j)}$.
L'estimation des scores est souvent faite par la méthode des moindres carrés (Chap.4 dans Ramsey and Silverman (2005)). Celle-ci nécessite d'avoir suffisamment de points dans les grilles.
Pour la dimension j, si l'on note $\tilde{\mathcal{T}}_j = \{t_1^{(j)}, \ldots, t_{k_j}^{(j)}\}$, le modèle suivant est considéré :

$$x_i^{(j)}(t_k^{(j)}) = \sum_{m=1}^{M_j} a_{i,m}^{(j)} \psi_m(t_k^{(j)}) + \epsilon_k$$

où $\mathbb{E}(\epsilon_s) = 0, \mathbb{E}(\epsilon_k^2) = \sigma^2 < \infty$, pour $k = 1, \ldots, k_j$.
Ainsi, soit le vecteur $a_i^{(j)} = \begin{pmatrix} a_{i,1}^{(j)} & \ldots & a_{i,M_j}^{(j)} \end{pmatrix}^\top$, son estimation est faite par la minimisation de :

$$\min_{b=(b_1,\ldots,b_M)^\top \in \mathbb{R}^{M_j}} \sum_{k=1}^{k_j} \left(x_i^{(j)}(t_k^{(j)}) - \sum_{m=1}^{M_j} b_m \psi_m(t_k^{(j)}) \right)^2 .$$

Cette approche est pratique parce qu'elle se met en œuvre rapidement et qu'elle peut facilement s'étendre au cas multivarié.
À noter que dans ce contexte, une estimation de l'opérateur de covariance s'écrit :

$$\hat{\Gamma}_{k,l}(s,t) = (\psi^{(k)}(s))^\top \frac{1}{n-1} \sum_{i=1}^{n} a_i^{(k)} (a_i^{(l)})^\top \psi^{(l)}(t). \tag{1.4}$$

où $\psi^{(k)} = \begin{pmatrix} \psi_1^{(k)} & \ldots & \psi_{M_k}^{(k)} \end{pmatrix}^\top$, $t \in \mathcal{T}_s$, $s \in \mathcal{T}_k$, $k = 1, \ldots, p$ et $s = 1, \ldots, p$.

Lorsque $\mathcal{T}_j \subset \mathbb{R}, j \in \{1, \ldots, p\}$ le vecteur $\psi^{(j)}$ est composé de fonctions univariées. Les systèmes de fonctions les plus utilisés dans ce cadre sont les bases Fourier et les B-splines, suivant la périodicité des données (Chap.3 dans Ramsey and Silverman (2005)). Elles sont brièvement exposées ci-dessous.

Les fonctions composant la base de Fourier ont les formes génériques suivantes :

$$\psi_{2m}^{(j)}(t) = \sin(m\omega t)$$
$$\psi_{2m+1}^{(j)}(t) = \cos(m\omega t)$$

où $\omega \neq 0, m \geq 2, \psi_1^{(j)}(t) = 1$ et $t \in \mathcal{T}_j$. Ces fonctions sont $2\pi/\omega$ périodiques et orthogonales entre elles

$$\langle \psi_m^{(j)}, \psi_k^{(j)} \rangle_{L_2(\mathcal{T}_j)} = 0$$

pour $k \neq m$.
Pour les images ($\mathcal{T}_j \subset \mathbb{R}^2$), une implémentation de la base en cosinus discrète est proposée dans le package Happ (2017). Cette dernière est proche des bases de Fourier dans la mesure où la transformation discrète en cosinus est une variante de la transformée de Fourier rapide (FFT).

Le système B-splines (*Basis-splines*) est composé de fonctions dites splines. Une spline est une fonction polynomiale par intervalles. Par exemple, soient $r_0, r_1, \ldots, r_K \in \mathcal{T}_j \subset \mathbb{R}$, les bornes de ces intervalles et $\mathcal{P}_k$ les fonctions polynomiales $[r_{k-1}, r_k) \to \mathbb{R}$, $k = 1, \ldots, K$:

$$\mathcal{T}_j = [r_0, r_1) \cup [r_1, r_2) \cup \ldots \cup [r_{K-1}, r_K) \cup [r_K].$$

La fonction $\mathcal{S}$ suivante est dite spline

$$\mathcal{S}(t) = \sum_{k=0}^{K-2} \mathcal{P}_k(t)\mathbb{I}_{[r_k, r_{k+1})}(t) + \mathcal{P}_K(t)\mathbb{I}_{[r_{K-1}, r_K]}(t),$$

où $\mathbb{I}$ est la fonction indicatrice.
Les points $r_0, \ldots, r_K$ sont appelés point de ruptures (*breakpoints*) ou nœuds (*knots*). L'ordre d'une spline est $dg + 1$, où dg est le plus haut degré des fonctions polynomiales ($\mathcal{P}_k$, $k = 0, \ldots, K$). Des détails sur la construction de la base splines peuvent être retrouvés dans De Boor (1972) et Ramsey and Silverman (2005)(Chap.3).

À des fins d'illustration, la Figure 1.1 présente la reconstruction d'une fonction univarié à l'aide de cette dernière base. Notons qu'il existe aussi des implémentations de *B-splines* pour les images ($\mathcal{T}_j \subset \mathbb{R}^2$) comme l'illustre la Figure 1.2. Cette figure est construite à partir de la base de données *Digit Recognizer*[1]. Chaque image est observée sur une grille de 28×28 pixels. La Figure 1.2 représente la reconstruction fonctionnelle de quatre images de "zéro" par une base spline à deux dimensions (les implémentations sont faites à l'aide de Happ (2017)). Quatre nombres de fonctions dans la base sont testés : 4 , 16 et 28, ici respectivement représentés par (b), (c) et (d).

1. https://www.kaggle.com/competitions/digit-recognizer/data

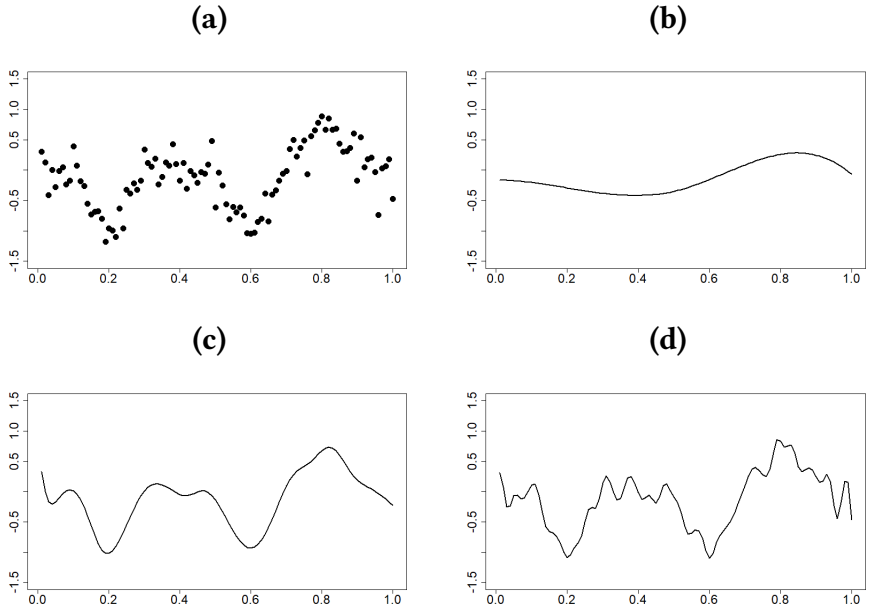

Figure 1.1 – Reconstructions fonctionnelles des observations **(a)** avec 5, 20 et 50 fonctions splines dans la base, respectivement représentée par **(b)**, **(c)** et **(d)**.

La Figure 1.2 montre clairement qu'un petit nombre (en l'occurrence 5) ne permet pas de reconstruire de manière satisfaisante les données, tandis que 28 fonctions en fournissent une reconstruction quasi parfaite. Un bon nombre de fonctions dans la base doit permettre une représentation satisfaisante des données, sans être trop grand. En effet, lorsque des erreurs de mesures sont présentes dans les observations, un grand nombre de fonctions conduit à inclure le bruit dans la modélisation. Cela est plus visible dans la Figure 1.1, représentant un cas avec des fonctions univariées.

Une fois les observations reconstituées en fonctions, l'analyse de celles-ci n'est pas triviale, en raison principalement des colinéarités et de la dimension souvent infinie des espaces de fonctions. Une méthode reconnue pour traiter cette problématique est l'analyse en composante principale (ACP) fonctionnelle.

1.2 Analyse en composante principale fonctionnelle

Comme dans le cadre multivarié classique, l'ACP fonctionnelle sert principalement deux buts : une interprétation des principaux modes de variations des données et la réduction de dimension. Les concepts clés de celle-ci proviennent des travaux de Karhunen (1946) et Loève (1946). Au fil des années, plusieurs études théoriques de l'ACP pour données fonctionnelles ont vu le jour. On peut se référer par exemple aux travaux suivants : Saporta (1981), Dauxois et al. (1982), Besse and Ramsay (1986), Hall et al. (2001), Ramsey and Silverman (2005), Chen and Müller (2012), Jacques and Preda (2014) et Happ and Greven (2018).

L'ACP fonctionnelle vise à trouver la base orthogonale de fonctions permettant une représentation optimale des données. Cette base est optimale dans la mesure où elle résume

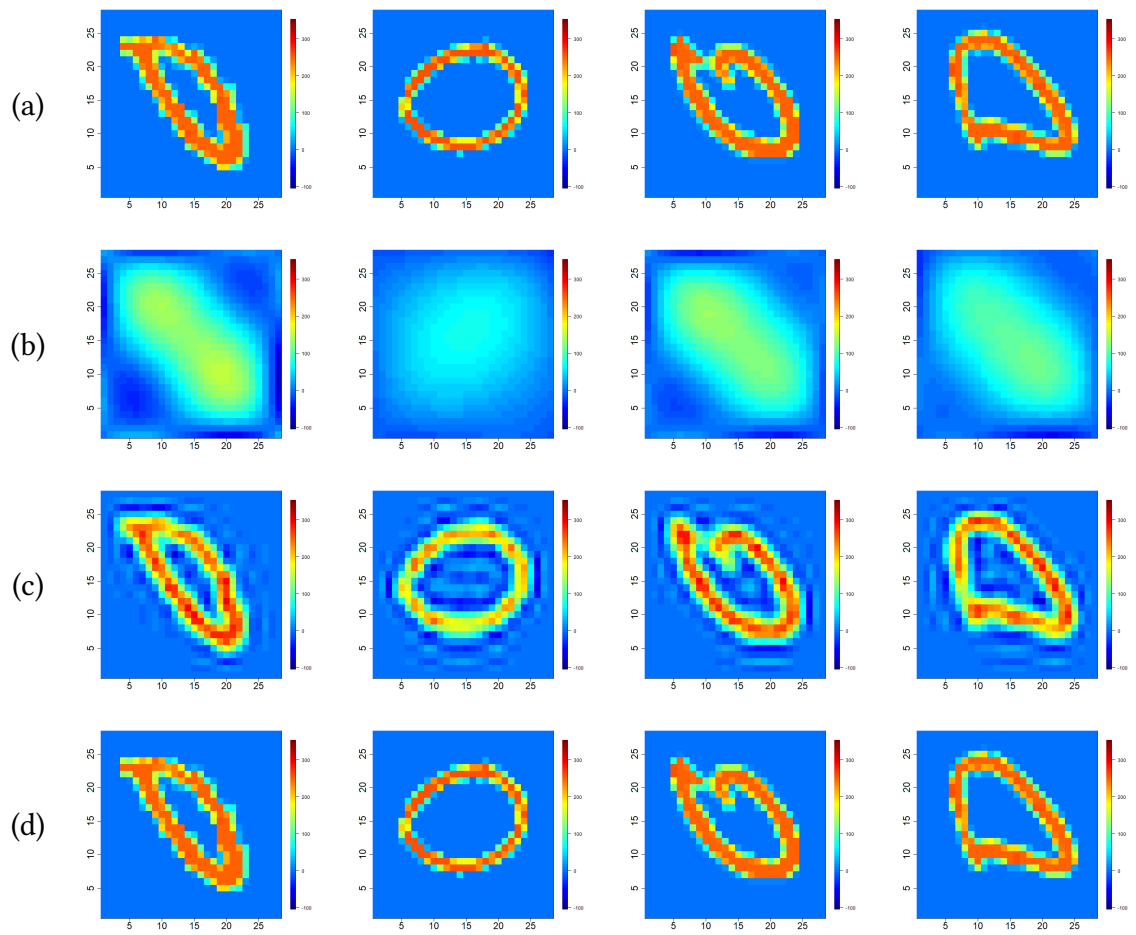

Figure 1.2 – Reconstructions fonctionnelles des observations en (a) à l'aide de B-Splines composée de 4, 16 et 28 fonctions respectivement représentées par (b), (c) et (d).

au mieux la variance des données.

Cette section présente l'ACP sur une variable aléatoire fonctionnelle X selon qu'elle est univariée (FPCA) ou multivariée (MFPCA).

1.2.1 FPCA

L'objectif du FPCA (Chap.8 dans Ramsey and Silverman (2005)) est de trouver les fonctions $\phi_1, \phi_2, \ldots$ qui permettent de résumer au mieux la variance des données.
Elles peuvent être cherchées de manière itérative.

- $\phi_1 = \arg\max_{f \in \mathcal{H}, \|f\|_{\mathcal{H}}=1} \mathbb{E}\left(\langle X, f\rangle_{\mathcal{H}}^2\right)$
- $\phi_2 = \arg\max_{f \in \mathcal{H}, \|f\|_{\mathcal{H}}=1} \mathbb{E}\left(\langle X, f\rangle_{\mathcal{H}}^2\right)$, avec $\langle \phi_1, \phi_2\rangle_{\mathcal{H}} = 0$

  ...
- $\phi_l = \arg\max_{f \in \mathcal{H}, \|f\|_{\mathcal{H}}=1} \mathbb{E}\left(\langle X, f\rangle_{\mathcal{H}}^2\right), \langle \phi_l, \phi_j\rangle_{\mathcal{H}} = 0 \; j = 1, \ldots, l-1$

  ...

La variance de la projection de X sur les fonctions $\phi_1, \phi_2, \ldots$ est maximale. Ces fonctions permettent d'identifier les modes de variations les plus forts et les plus importants de la variable fonctionnelle. Les deux contraintes : la norme unitaire et l'orthogonalité, servent à la bonne définition du problème. Elles assurent, d'abord, que cette variance ne soit pas arbitrairement élevée, ensuite, que les fonctions fournissent des informations non redondantes (Ramsey and Silverman, 2005).

On peut montrer (voir Théorème 3.2 dans Horváth and Kokoszka (2012)) que $\phi_1, \phi_2, \ldots$ sont les fonctions propres de l'opérateur de covariance Γ.

$$\Gamma\phi_k(t) = \lambda_k\phi_k(t),\ t \in \mathcal{T},\ k = 1, 2, \ldots \tag{1.5}$$

où

$$\Gamma\phi_k(t) = \int_{\mathcal{T}} \Gamma(t, s)\phi_k(s)ds$$

et $\Gamma(s, t) = \text{Cov}(X(s), X(t))$. De plus, soient $\xi_k = \langle X, \phi_k\rangle_{\mathcal{H}}$, $k = 1, \ldots$ les projections de X sur les fonctions propres de Γ, alors :

$$\lambda_1 \geq \lambda_2 \geq \ldots \geq 0,$$

où $\lambda_k = \mathbb{E}(\xi_k^2)$.
Ainsi, la variable aléatoire fonctionnelle X admet une expansion de Karhunen-Loève

$$X(t) = \sum_{m\geq 1} \xi_m\phi_m(t),$$

où $t \in \mathcal{T}$ et $\{\phi_k\}_{k\geq 1}$ est une base dans $\mathcal{H}$.
Trouver les fonctions $\phi_1, \phi_2, \ldots$ est alors équivalent à résoudre le problème suivant :

- $\phi_1 = \arg\max_{f\in\mathcal{H}, ||f||_{\mathcal{H}}=1} \langle\Gamma f, f\rangle_{\mathcal{H}}$
- $\phi_2 = \arg\max_{f\in\mathcal{H}, ||f||_{\mathcal{H}}=1} \langle\Gamma f, f\rangle_{\mathcal{H}}$, avec $\langle\phi_1, \phi_2\rangle_{\mathcal{H}} = 0$

  $\ldots$
- $\phi_l = \arg\max_{f\in\mathcal{H}, ||f||_{\mathcal{H}}=1} \langle\Gamma f, f\rangle_{\mathcal{H}}$, $\langle\phi_l, \phi_j\rangle_{\mathcal{H}} = 0\ j = 1, \ldots, l-1$

  $\ldots$

Dans les faits, comme déjà discuté dans la section précédente, les réalisations de $x_1, \ldots, x_n$ de X sont souvent représentées dans un système de fonctions $\psi = (\psi_1, \ldots, \psi_M)^\top$:

$$x_i(t) = a_i^\top\psi(t)$$

où $a_i = \begin{pmatrix} a_{i,1} & \ldots & a_{i,M}\end{pmatrix}^\top \in \mathbb{R}^M$ et $i = 1, \ldots, n$.
Alors les fonctions FPCA peuvent-être estimées à l'aide de l'approximation de l'opérateur de covariance Γ (voir (1.4))

$$\hat{\Gamma}(s, t) = \frac{1}{n-1}\psi^\top(s)\mathbf{A}^\top\mathbf{A}\psi(t)\ \ s, t \in \mathcal{T}$$

où

$$\mathbf{A} = \begin{pmatrix} a_{1,1} & \dots & a_{1,M} \\ \vdots & \dots & \vdots \\ a_{l,1} & \dots & a_{l,M} \\ \vdots & \dots & \vdots \\ a_{n,1} & \dots & a_{n,M} \end{pmatrix}.$$

Supposons que les fonctions propres de l'opérateur de covariance admettent une expansion dans la base de ψ

$$\phi_m(t) = b_m^\top \psi(t) \tag{1.6}$$

où $b_m \in \mathbb{R}^M$, $m = 1, 2, \dots$, alors une estimation de b_m est

$$\frac{1}{n-1}\mathbf{A}^\top \mathbf{A}\mathbf{W}\hat{b}_m = \lambda_m \hat{b}_m \tag{1.7}$$

où $\hat{b}_m \mathbf{W} b_m = 1$, $\hat{b}_k \mathbf{W} \hat{b}_m = 0$ pour $k \neq m$. La matrice $\mathbf{W}$ est la matrice des produits scalaires des fonctions dans la base

$$\mathbf{W} = \begin{pmatrix} \langle \psi_1, \psi_1 \rangle_{\mathcal{H}} & \dots & \langle \psi_1, \psi_M \rangle_{\mathcal{H}} \\ \vdots & \dots & \vdots \\ \langle \psi_l, \psi_1 \rangle_{\mathcal{H}} & \dots & \langle \psi_l, \psi_M \rangle_{\mathcal{H}} \\ \vdots & \dots & \vdots \\ \langle \psi_M, \psi_1 \rangle_{\mathcal{H}} & \dots & \langle \psi_M, \psi_M \rangle_{\mathcal{H}} \end{pmatrix}.$$

La section 8.4.2 dans Ramsey and Silverman (2005) donne des détails sur ces calculs et les procédures d'estimation des vecteurs b_m, $m = 1, 2, \dots$

Des extensions de l'approche FPCA pour les images ($\mathcal{T} \subset \mathbb{R}^2$) ont été proposées, par exemple, dans Zhou and Pan (2014) et Locantore et al. (1999). L'article de synthèse Shang (2014) offre une vue d'ensemble des travaux sur la littérature relative à la FPCA.

Pour illustration, revenons à l'exemple précédent. Considérons les 4132 exemples de "zéro" dans la base de données *Digit Recognizer*. L'analyse principale est faite sur ces données, en utilisant une base de 16 fonctions splines (voir la Figure 1.2 (c)). Les quatre premières fonctions propres représentées dans la Figure 1.3 expliquent 62.71% de la variance des données.

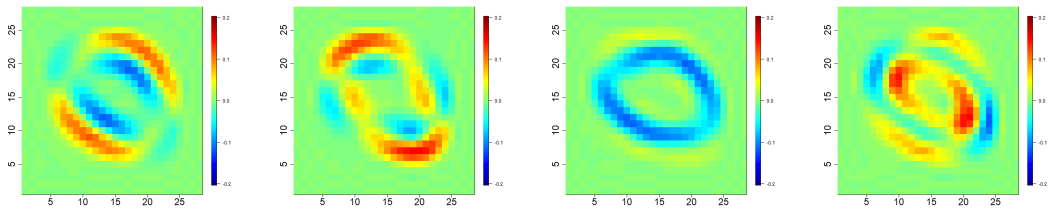

Figure 1.3 – Les quatre premières fonctions de l'ACP sur *Digit Recognizer*

1.2.2 MFPCA

Le concept de "données fonctionnelles multivariées" peut désigner une variété de données fonctionnelles. Elles peuvent être définies sur le même domaine ou sur des domaines différents. Au sein de ces définitions, des nuances existent. En effet, lorsqu'elles sont dans le premier cas, il peut s'agir d'une même quantité mesurée suivant plusieurs *conditions*. Ce type de variable est connu sous l'appellation de variable fonctionnelle répétée. Dans ce cadre, dans Chen and Müller (2012) une analyse en composante principale est proposée. Ce travail rentre dans le cadre plus général des données longitudinales (voir par exemple Yao et al. (2005), Greven et al. (2011), etc.). Ici, nous faisons le choix de nous concentrer sur le cas de variables fonctionnelles multivariées définies sur des domaines différents.

L'ACP présentée dans Happ and Greven (2018) est proposée pour des variables fonctionnelles vérifiant la définition du point **b** de la Section 1.1. Comme pour le cas univarié, l'objectif est de trouver les fonctions propres de l'opérateur de covariance Γ. On rappelle qu'ici Γ est défini comme une matrice de fonctions (voir l'équation (1.1)).

En utilisant la FPCA, l'expansion de Karhunen-Loève de chaque dimension $j = 1, \dots, p$ de X donne que :

$$X^{(j)} = \sum_{j=1}^{M_j} \xi_m^{(j)} \tilde{\phi}_m^{(j)}$$

où $\{\tilde{\phi}_m^{(j)}\}_{m=1}^{M_j}$ sont les M_j-fonctions propres (non nulles) de $\Gamma^{(j)}(s,t) = \mathbb{E}(X^{(j)}(s)X^{(j)}(t))$ $s, t \in \mathcal{T}_j$. Supposons que les quantités M_j soient finies – c'est-à-dire que $M_j \in \mathbb{N}^*$, pour $j = 1, \dots, p$– les auteures dans Happ and Greven (2018) montrent (Proposition 5) que les fonctions propres de Γ, notées $\{\phi_k\}_{k\geq 1}$, peuvent s'écrire comme une combinaison linéaire des fonctions obtenues par la FPCA.

$$\phi_m^{(j)}(t) = \sum_{n=1}^{M_j} c_{m,n}^{(j)} \tilde{\phi}_n^{(j)}(t)$$

où $c_{m,n}^j \in \mathbb{R}$. Plus précisément, $c_m^{(j)} = \begin{pmatrix} c_{m,1}^{(j)} & c_{m,2}^{(j)} & \dots & c_{m,M_j}^{(j)} \end{pmatrix}^\top$ et $c_m = \begin{pmatrix} c_m^{(1)} & \dots & c_m^{(p)} \end{pmatrix}^\top$ correspond au m-ième vecteur propre de

$$\mathbf{Z} = \begin{pmatrix} Z^{(1,1)} & \dots & Z^{(1,l)} & \dots & Z^{(1,p)} \\ \vdots & \dots & & \dots & \vdots \\ Z^{(j,1)} & \dots & Z^{(j,l)} & \dots & Z^{(j,p)} \\ \vdots & \dots & & \dots & \vdots \\ Z^{(p,1)} & \dots & Z^{(p,l)} & \dots & Z^{(p,p)} \end{pmatrix}$$

où

$$Z^{(j,l)} = \begin{pmatrix} \mathrm{Cov}(\xi_1^{(j)}, \xi_1^{(k)}) & \dots & \mathrm{Cov}(\xi_1^{(j)}, \xi_u^{(k)}) & \dots & \mathrm{Cov}(\xi_1^{(j)}, \xi_{M_k}^{(k)}) \\ \vdots & \dots & \vdots & \dots & \vdots \\ \mathrm{Cov}(\xi_v^{(j)}, \xi_1^{(k)}) & \dots & \mathrm{Cov}(\xi_v^{(j)}, \xi_u^{(k)}) & \dots & \mathrm{Cov}(\xi_v^{(j)}, \xi_{M_k}^{(k)}) \\ \vdots & \dots & \vdots & \dots & \vdots \\ \mathrm{Cov}(\xi_{M_j}^{(j)}, \xi_1^{(k)}) & \dots & \mathrm{Cov}(\xi_{M_j}^{(j)}, \xi_u^{(k)}) & \dots & \mathrm{Cov}(\xi_{M_j}^{(j)}, \xi_{M_k}^{(k)}) \end{pmatrix}.$$

Lorsque toutes les dimensions de la variable fonctionnelle sont définies sur le même domaine, les auteurs dans Jacques and Preda (2014) proposent d'utiliser des bases splines pour estimer les fonctions propres de l'opérateur de covariance de X. Les scores $\xi_1, \xi_2, \ldots$ (obtenue par la MFPCA) sont alors utilisés pour du *clustering* (classification non supervisée). Pour p (le nombre de dimensions) plus grand ou proche du nombre de réalisations (n), dans Hu and Yao (2020), les auteurs présentent une analyse en composante principale parcimonieuse.

L'ACP fonctionnelle sert à analyser la v.a.f X. Il est possible que celle-ci soit associée à une variable d'intérêt Y (qui peut être scalaire ou fonctionnelle). L'analyse (ou approche) supervisée étudie le problème de la prédiction de Y à partir de X.

1.3 Approche supervisée

Ici, nous nous limitons au cas d'une variable aléatoire scalaire Y. Les modèles principalement discutés dans cette section sont des modèles paramétriques, plus facilement interprétables que les non paramétriques. Pour une vision globale des contributions sur ces derniers, on peut se référer, notamment, aux travaux de Ferraty and Vieu (2006) et Kokoszka and Reimherr (2017).

La variable à expliquer peut être soit quantitative, soit catégorielle. Dans le premier cas, Y peut prendre ses valeurs dans tout l'ensemble $\mathbb{R}$ sans restriction. Mais, lorsqu'elle est catégorielle à K modalités, Y prend ses valeurs dans l'ensemble $\{1, \ldots, K\}$. Ainsi, cette partie, qui porte sur le problème de la prédiction de Y à l'aide de X, se divise en deux sections. D'abord, la régression, lorsque Y est quantitative. Ensuite, la classification, lorsque Y est catégorielle.

1.3.1 La régression linéaire

Le modèle de régression linéaire suppose que l'espérance conditionnelle de Y sachant X existe et qu'elle est linéaire.

$$\mathbb{E}(Y|X) = \langle X, \beta \rangle_{\mathcal{H}} \tag{1.8}$$

Peut-être grâce à son potentiel interprétatif, une large littérature s'est concentrée sur cette modélisation. Le principal cadre d'investigation considère la v.a.f univariée, voir par exemple Morris (2015). Quelques travaux, récemment, ont exploré le cas des variables explicatives fonctionnelles multivariées. Une majorité d'entre eux font l'hypothèse d'un domaine de définition commun pour toutes les dimensions (Koner and Staicu, 2023). Cette partie propose un bref aperçu de quelques-uns de ces travaux.

Supposons que Y est de moyenne nulle, le modèle linéaire fonctionnel est

$$Y = \langle X, \beta \rangle_{\mathcal{H}} + \epsilon \tag{1.9}$$

où ϵ désigne le terme résiduel de moyenne nulle. Il est supposé non corrélé à X et de variance finie $\mathbb{E}(\epsilon^2) = \sigma^2 < \infty$.

L'estimation directe par le critère des moindres carrés de la fonction coefficient (β) est un problème inverse mal posé : Γ n'est pas inversible (Cardot et al. (1999)).
Les techniques couramment utilisées approchent β à l'aide d'une combinaison linéaire de fonctions dans $\mathcal{H} : g_1, \ldots, g_M$.

$$\hat{\beta}^{(j)}(t) = \sum_{k=1}^{M} b_k g_k^{(j)}(t), \ t \in \mathcal{T}_j \tag{1.10}$$

$j = 1, \ldots, p$.

Une première idée serait de choisir la base ψ qui sert pour la reconstruction des données fonctionnelles. Cependant, dans ce cadre, le critère des moindres carrés pour l'estimation de $b = \big(b_1, \ldots, b_k\big)^\top$, surtout lorsqu'il y a un grand nombre de fonctions de base, mène à des problèmes de multicolinéarité et de sur-apprentissage (Aguilera et al., 2010). La régression pénalisée (PR), la régression en composantes principales (PCR) et la régression des moindres carrés partiels (PLS) sont utilisées pour contourner cette difficulté.

La régression pénalisée : Comme mentionné ci-dessus, l'utilisation d'une base arbitraire ψ peut conduire à des difficultés liées à la grande dimension et une multicolinéarité. La régression pénalisée rajoute des contraintes lors de la minimisation du critère des moindres carrés afin d'estimer au mieux la fonction coefficient. Le cadre le plus largement étudié est : X univarié, c.-à-d. $p = 1$ et $\mathcal{H} = L_2(\mathcal{T})$. Pour une vue d'ensemble, on peut se reporter par exemple à la Section 2.1 de l'article de Reiss et al. (2017).
L'objectif de la régression pénalisée revient, lorsque la base ψ est utilisée, à trouver le vecteur b tel que :

$$\hat{\beta} = \sum_{k=1}^{M} b_k \psi_k(t), \ t \in \mathcal{T}, \text{ tel que } \mathcal{P}(\hat{\beta}) \leq s \tag{1.11}$$

où $\mathcal{P} : \mathcal{H} \mapsto \mathbb{R}$ est la fonction pénalité et s un seuil à fixer. De cette façon, le but est de trouver un compromis entre la performance du modèle et certaines propriétés d'intérêt caractérisées par $\mathcal{P}$. Par exemple, on peut citer les deux pénalités ci-dessous.

- La pénalité $\mathcal{P}(\hat{\beta}) = \|\hat{\beta}''\|_{L_2} = \sum_{k=1}^{M} |b_k| \|\psi_k''\|_{L_2}$ exige d'avoir une fonction coefficient suffisamment lisse suivant le seuil s (voir par exemple Sørensen et al. (2013)).
- L'approche *sparse* $\mathcal{P}(\hat{\beta}) = \sum_{k=1}^{M} |b_k|$ a pour objectif de maximiser suivant le seuil s les parties de l'intervalle $\mathcal{T}$ où la fonction β est nulle (voir par exemple Lee and Park (2012)).

Ces deux types de pénalités n'ont pas les mêmes implications. En effet, si l'on note $\hat{\beta}^r$ la r-iéme dérivée de $\hat{\beta}$, alors la première pénalité dépend de la dérivée $r = 2$ et la seconde de la dérivée $r = 0$. La méthode proposée dans James et al. (2009), sans utiliser l'expansion en base, permet d'agir simultanément sur les $r = 0, 1, 2, \ldots$ dérivées. De cette façon, lorsqu'on se limite à $r \leq 2$, la fonction estimée est recherchée de manière à avoir dans certaines parties de son domaine de définition $\mathcal{T}$ les propriétés suivantes :

- nulle

$$\{t \in \mathcal{T}, \beta^0(t) = 0\}$$

— constante
$$\{t \in \mathcal{T}, \beta^1(t) = 0\}$$
— linéaire
$$\{t \in \mathcal{T}, \beta^2(t) = 0\}.$$

La régression en composantes principales : Lorsque les M-premières fonctions obtenues par la MFPCA ($\phi_1, \ldots, \phi_M$) sont utilisées pour approximer β, l'équation (1.9) revient à :
$$Y = \sum_{k=1}^{M} b_k \xi_k + \epsilon \tag{1.12}$$
où $\xi_k = \langle X, \phi_k \rangle_{\mathcal{H}}$. Les propriétés asymptotiques de β, sous la régression sur les composantes principales, ont été étudiées dans Cardot et al. (1999) lorsque la variable explicative est univariée fonctionnelle. Dans Reiss and Ogden (2007), les auteurs proposent deux méthodologies PCR fonctionnelles (FPCR) intégrant des termes de pénalités. Ces méthodes servent à une meilleure interprétation de la fonction coefficient et sont présentées dans le cadre univarié fonctionnel. Il existe dans la littérature d'autres utilisations de la base de fonctions obtenue par la FPCA dans le cadre de la régression. Par exemple, celle-ci est utilisée dans Goldsmith et al. (2011), où les auteurs proposent une approche flexible pour des données fonctionnelles multivariées sur des grilles d'observations irrégulières. Par ailleurs, dans Hall and Horowitz (2007), les auteurs présentent une étude des taux de convergences de l'estimateur.

La régression des moindres carrés partiels : L'approche PLS pour les données fonctionnelles univariée introduite dans Preda and Saporta (2002) est itérative. Elle cherche de manière renouvelée la fonction w qui maximise le critère de Tucker
$$w = \underset{f \in \mathcal{H}, ||f||_{\mathcal{H}}=1}{\arg\max} \ \mathrm{Cov}^2(\langle X, f \rangle_{\mathcal{H}}, Y). \tag{1.13}$$
L'algorithme PLS est donné par :
— Étape 0 : Soit $X_0 = X$ et $Y_0 = Y$.
— Étape h, $h \geq 1$:
  — Trouver la fonction w_h qui maximise (1.13), avec $X = X_{h-1}$ et $Y = Y_{h-1}$. La h-ième composante PLS est :
  $$\zeta_h = \langle X_{h-1}, w_h \rangle_{\mathcal{H}},$$
  — Déterminer les résidus X_h and Y_h de la régression linéaire de X_{h-1} et Y_{h-1} sur ξ_h,
  $$X_h = X_{h-1} - \rho_h \zeta_h,$$
  $$Y_h = Y_{h-1} - c_h \zeta_h,$$
  où $\rho_h = \dfrac{\mathbb{E}(X_{h-1}\zeta_h)}{\mathbb{E}(\zeta_h^2)} \in \mathcal{H}$ et $c_h = \dfrac{\mathbb{E}(Y_{h-1}\zeta_h)}{\mathbb{E}(\zeta_h^2)} \in \mathbb{R}$.
— Aller à l'étape suivante ($h = h + 1$).

Lorsque les M composantes PLS sont déterminées, on peut montrer que la fonction coefficient est estimée par β_M

$$\beta_M = \sum_{k=1}^{M} c_k v_k.$$

où $c_k \in \mathbb{R}$ pour $k = 1, \ldots, M$ et les fonctions $v_1, \ldots, v_M$ appartiennent à l'espace engendré par les fonctions $w_1, \ldots, w_M$ (Aguilera et al., 2010).
Depuis l'introduction de la régression PLS sur les données fonctionnelles (FPLS), cette approche a été au cœur de nombreux développements et contributions. Toujours dans le cadre univarié, les travaux dans Aguilera et al. (2010) ont démontré que FPLS peut être mis en relation avec la procédure PLS classique à l'aide de l'expansion dans une base de fonctions. Dans Delaigle and Hall (2012b), les auteurs en proposent une version alternative (non itérative) et en dérivent les propriétés asymptotiques. Plus récemment, les auteurs dans Beyaztas and Shang (2022) proposent une extension du résultat présenté dans Aguilera et al. (2010) pour les données fonctionnelles multivariées définies sur un même domaine.

Les méthodes PCR et PLS sont réputées pour l'efficacité de leur algorithme d'estimation et l'interprétabilité des résultats. Comme mentionné dans Jong (1993), dans le cadre classique (dimension finie), et dans Aguilera et al. (2010), dans le cadre fonctionnel, pour un nombre fixé de composantes, la régression PLS fournit un meilleur modèle (en termes de critère de la somme carrée des erreurs) que PCR. Des expériences numériques confirment ces résultats pour la régression avec des données fonctionnelles univariées (Delaigle and Hall (2012b), Guan et al. (2022)). Dans Febrero-Bande et al. (2017), une étude comparée détaillée des deux méthodes est proposée lorsque $\mathcal{H} = L_2(\mathcal{T})$.

1.3.2 Classification

L'analyse discriminante et la régression logistique sont des méthodes linéaires couramment utilisées pour la classification dans le cadre multivarié classique. Elles admettent des généralisations pour les données fonctionnelles. La plupart d'entre elles se concentrent, comme dans la partie précédente, sur le cas d'une variable fonctionnelle univariée.

Modèle logistique : Le modèle logistique est donné par :

$$\log \frac{\mathbb{P}(Y = k|X)}{1 - \sum_{l=1}^{K} \mathbb{P}(Y = l|X)} = \alpha_k + \langle X, \beta_k \rangle_{\mathcal{H}} \quad k = 1, \ldots, L$$

De la même manière que pour le cas de la régression linéaire, une approximation directe des fonctions $\beta_k \in \mathcal{H}$, $k = 1, \ldots, K$ n'est pas faisable. Dans Escabias et al. (2004), les auteurs proposent deux approches basées sur l'ACP pour estimer la fonction coefficient, lorsque X est univariée et $K = 2$. La première est similaire au PCR dans la mesure où elle utilise les fonctions obtenues par la FPCA. Les scores $\{\xi_k\}_{k\geq 1}$ remplacent la variable X dans le modèle logistique. La deuxième approche est basée sur l'ACP classique. Elle est faite sur la matrice des scores obtenus après l'expansion dans une base de fonctions.
Dans Godwin (2013), le modèle proposé est une extension de la deuxième technique, lorsque X est multivariée et définie sur un même domaine.

L'approche PLS pour les données fonctionnelles a aussi été explorée dans ce cadre. Voir Escabias et al. (2007), où les auteurs étendent aux données fonctionnelles univariées la régression logistique PLS proposée dans Bastien et al. (2005).

Analyse discriminante linéaire : Lorsque la variable explicative est fonctionnelle, l'objectif de l'analyse discriminante linéaire peut se définir (Preda et al., 2007) par :

$$\beta_{\text{DA}} = \arg\max_{\beta \in \mathcal{H}} \frac{\mathbb{V}(\mathbb{E}(\langle X, \beta \rangle_{\mathcal{H}} | Y))}{\mathbb{V}(\langle X, \beta \rangle_{\mathcal{H}})} \tag{1.14}$$

où $\mathbb{V}(.)$ désigne la variance.
Les articles James and Hastie (2001) et Preda et al. (2007) étudient ce modèle pour $\mathcal{H} = L_2(\mathcal{T})$ et $\mathcal{T} \subset \mathbb{R}$.
Dans James and Hastie (2001), sous l'hypothèse gaussienne des scores obtenus par l'expansion en base des observations fonctionnelles, l'analyse discriminante linéaire est estimée à l'aide de l'algorithme EM (Dempster et al., 1977). La méthode est en particulier proposée pour des courbes irrégulièrement échantillonnées.
Dans Preda et al. (2007), lorsque $K = 2$, l'approche présentée utilise la régression PLS pour estimer β_{DA}. En effet, l'estimation de β_{DA} est un problème inverse mal posé et, via un recodage de la variable Y, peut se ramener à un problème de régression.

En discrétisant le critère ci-dessus, dans Gardner-Lubbe (2021), les auteurs proposent une analyse discriminante pour les données fonctionnelles multivariée définies sur le même domaine.

Arbres de décision : Des travaux récents s'intéressent à l'utilisation en FDA des arbres de décision (et parfois pour de la régression). L'étape de partitionnement des nœuds est centrale dans la construction de l'arbre. Dans le cadre classique ($\mathrm{x} = (\mathrm{x}^1, \ldots, \mathrm{x}^p)^\top \in \mathbb{R}^p$), par exemple pour l'arbre CART (Breiman, 1984), l'objectif est de trouver la variable x^j et le seuil c_j qui permettent d'obtenir des nœuds de plus en plus purs après chaque étape ($\mathrm{x}^j \leq c_j$ et $\mathrm{x}^j > c_j$). Pour des variables explicatives fonctionnelles univariées une traduction directe de cette procédure équivaudrait à chercher $t_j \in \mathcal{T}$ et le seuil c_j tel que $X(t_j) \leq c_j$ et $X(t_j) > c_j$. Autrement dit, cela reviendrait à tester une infinité d'éléments $t \in \mathcal{T}_j$ ($\mathcal{T}_j$ est un intervalle compact).

Cette extension n'étant pas implémentable, les arbres pour les données fonctionnelles proposent d'autres méthodes pour le partitionnement.

- **Réduction de dimension :** Dans Maturo and Verde (2022), les auteurs proposent d'utiliser les scores obtenus par la FPCA pour classer des courbes univariées à l'aide d'un arbre de décision. Il est possible d'utiliser d'autres bases de fonctions, par exemple dans El Haouij et al. (2019) les auteurs présentent un modèle de forêt aléatoire (ensemble d'arbre de décision) fonctionnel qui prend en entrée les scores de la variable sur une base d'ondelettes.
- **Distance :** La méthode de classification binaire ($K = 2$) proposée dans Möller and Gertheiss (2018) prend en entrée les données fonctionnelles multivariées. Le partitionnement est fait à l'aide de la proximité relative aux moyennes des classes. L'al-

gorithme utilisé est illustré dans la Figure 1.4, les deux classes sont notées "A" et "B".

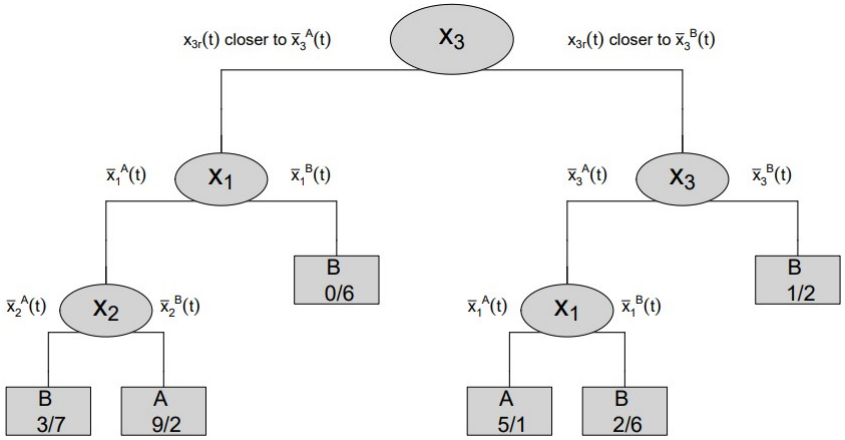

Figure 1.4 – Figure 1 dans Möller and Gertheiss (2018)

— **Partitionnement fonctionnel :** Très récemment dans Belli and Vantini (2022), l'approche présentée exploite l'aspect fonctionnel des données. Elle est résumée ci-dessous.
Supposons que $\mathcal{H} = L_2(\mathcal{T})$ (cas univarié), soient les fonctions $w : \mathcal{T} \to \mathbb{R}$ et $f(., w) : \mathcal{H} \to \mathbb{R}$.
Les règles de partitionnement sont données par :
  — $\mathcal{R}_1^f(w, s) = \{x | f(x, w) \leq s\}$
  — $\mathcal{R}_1^f(w, s) = \{x | f(x, w) > s\}$

Les fonctions f, w et le seuil s sont sélectionnés pour le partitionnement lorsqu'elles minimisent le critère

$$\mathcal{L}(\mathcal{R}_1^f(w, s)) + \mathcal{L}(\mathcal{R}_2^f(w, s)) \tag{1.15}$$

où $\mathcal{L}$ est la fonction coût (*loss function*). Elle est, par exemple, l'indice de Gini pour la classification et la moyenne quadratique des erreurs pour la régression.
Trouver les fonctions f, w et le seuil s est un problème non convexe. La méthode proposée par les auteurs consiste à fixer f et chercher les fonctions poids w à l'aide de la régression fonctionnelle pénalisée.

L'approche proposée dans Blanquero et al. (2023) utilise aussi des procédures d'optimisation pour le partitionnement des nœuds. Cette méthode, proposée pour la régression, prend compte de l'aspect fonctionnel des données. Elle est basée sur les arbres de décision optimaux (*Optimal decision tree*), qui, contrairement aux arbres classiques, permettent de trouver les divisions le long de l'arbre en une seule fois (Carrizosa Priego et al., 2021).

Première partie

Données fonctionnelles multivariées sur plusieurs domaines

Chapitre 2

L'approche PLS pour la classification de données fonctionnelles multivariées

Ce chapitre explore la classification (binaire) des données fonctionnelles multivariées avec des domaines différents. Dans ce cadre, nous présentons la régression des moindres carrés partiels sur les données fonctionnelles multivariées (MFPLS). Nous montrons que les composantes MFPLS peuvent être obtenues en utilisant la régression des moindres carrés partiels sur des données fonctionnelles univariées (FPLS). Cela offre une nouvelle façon de présenter l'algorithme PLS pour les données fonctionnelles multivariées. Les études de simulations et les applications aux données réelles montrent les performances de MFPLS pour la classification et la régression.

2.1 Introduction

Dans de nombreux domaines, des données "haute fréquence" sont enregistrées en fonction du temps et de l'espace. En médecine par exemple, les praticiens peuvent s'appuyer sur des données de séries temporelles (électroencéphalogramme, électrocardiogramme, etc.) ou d'images (fMRI, etc.) pour diagnostiquer l'état d'un patient. En finance, les marchés boursiers sont naturellement enregistrés de manière temporelle et à des localisations précises. L'analyse de telles données, en raison de leurs complexités (corrélation et grande dimension entre autres), requiert des techniques adaptées. Depuis les travaux pionniers de Ramsey and Silverman (2005), ces données sont dites fonctionnelles. L'analyse de données fonctionnelles (FDA) considère un échantillon comme les réalisations d'une variable aléatoire (X) prenant ses valeurs dans un espace de dimension infinie ($\mathcal{H}$). La variable X est souvent associée à un paramètre continu tel que le temps, les longueurs d'ondes ou le pourcentage d'un cycle. La FDA est de nos jours un sujet bien établi de recherches en statistique.

Pour éviter les problèmes liés à la grande dimension et la corrélation de ces données, les techniques de réduction de dimensions ont largement été utilisées. Parmi elles, la plus élémentaire est la sélection de caractéristiques sur les données (moyenne, variance, etc.), voir par exemple Saikhu et al. (2019), Javed et al. (2020). Cette technique dépend donc de connaissances apriori prodiguées par les experts. D'autres travaux se sont axés sur les techniques d'apprentissage profond (*deep learning*). Par exemple, les modèles *Long Short-Term Memory* ont prouvé leur efficacité pour la classification de séries temporelles (Hochreiter and Schmidhuber (1997), Karim et al. (2017), Karim et al. (2019)). Ils ont l'avantage d'être moins dépendants des connaissances apriori, mais ne conduisent généralement pas à des modèles interprétables. Une autre méthode consiste à traiter ces données comme des données fonctionnelles, c'est-à-dire des réalisations de variables aléatoires à valeur dans un espace de fonction. Dans ce cadre, les méthodologies les plus utilisées sont probablement celles fondées sur la construction de modèles latents tels que l'analyse/régression en composantes principales (PCA, PCR) (Ramsey and Silverman (2005), Jacques and Preda (2014), Escabias et al. (2004)) et les moindres carrés partiels (PLS) (Aguilera et al. (2010), Preda et al. (2007)). Ces modèles latents ne sont pas dépendants d'informations apriori et ils sont plus interprétables par rapport aux modèles de *deep-learning*.

Dans ce chapitre, par le prisme de l'analyse de données fonctionnelles, nous nous intéressons au problème de la prédiction d'une variable Y binaire à l'aide d'une variable explicative X fonctionnelle multivariée $X = (X^{(1)}, \ldots, X^{(p)})^\top$. Pour $j = 1, \ldots, p$, la composante $X^{(j)}$ est une variable fonctionnelle univariée : $X^{(j)} = \{X^{(j)}(t), t \in \mathcal{T}_j\}$, où $\mathcal{T}_j$ est un

ensemble compact d'index.

La classification supervisée des données fonctionnelles univariées ($p = 1$) a été la source de diverses contributions. Dans James and Hastie (2001), l'analyse discriminante linéaire multivariée (LDA) est étendue à des courbes irrégulièrement échantillonnées. Comme la maximisation de la variance intra-classes par rapport à la variance totale conduit à un problème mal posé, dans Preda et al. (2007), les auteurs proposent pour la classification une analyse discriminante fondée sur la régression PLS sur les données fonctionnelles univariées (FPLS). En utilisant le concept de profondeur (*depth*), dans López-Pintado and Romo (2006), des procédures robustes ont été introduites pour classer des données fonctionnelles. Les approches non paramétriques dans ce cadre ont également été étudiées, comme dans Ferraty and Vieu (2006) et Galeano et al. (2015). Elles utilisent les distances ou les mesures de similarités. L'utilisation des arbres de décisions pour la classification des données fonctionnelles est assez récente. L'article de Maturo and Verde (2022) présente un arbre de décision construit à partir des scores de l'ACP (Analyse en composante principale). Dans Möller and Gertheiss (2018), les distances entre les courbes sont exploitées pour la construction d'un arbre de décision pour la classification binaire.

Dans le cadre des données fonctionnelles multivariées, la classification supervisée est principalement étudiée lorsque tous les domaines $\mathcal{T}_j$ sont identiques. Autrement dit, les p-composants de X sont définis sur le même domaine $\mathcal{T}_j = [0, T]$, où $T > 0$ et $j = 1, \ldots, p$. Sous cette hypothèse, le travail de Blanquero et al. (2019) introduit une méthode permettant une sélection optimale des points les plus informatifs dans l'intervalle $[0, T]$. Dans Górecki et al. (2015), les modèles de régression sont utilisés pour classer les données fonctionnelles multivariées. Récemment, les auteurs dans Gardner-Lubbe (2021) ont proposé une analyse discriminante linéaire. Les auteurs utilisent des techniques de discrétisation afin de maximiser le ratio-variance intra et extra groupe dans le cas fonctionnel.

La classification de données fonctionnelles multivariées lorsque les domaines $\mathcal{T}_j$ sont différents est rarement explorée. Ce cadre, pourtant plus flexible, permet l'utilisation simultanée de données de types différents (par exemple, séries temporelles et images). À notre connaissance, seuls les auteurs dans Golovkine et al. (2022) ont proposé une méthode de classification applicable au cadre supervisée. En effet, la méthode présentée repose sur un arbre pour la classification, à l'origine, non supervisée. Néanmoins, les auteurs démontrent la possibilité d'utiliser cette approche pour la classification supervisée. Leur méthode est fondée sur l'analyse en composante principale pour des données fonctionnelles multivariées définies en différents domaines (MFPCA) (Happ and Greven, 2018).

L'utilisation de MFPCA, comme de l'ACP ordinaire, entraîne certains problèmes non triviaux pour l'apprentissage supervisé. Ces problèmes reposent principalement sur le nombre et la sélection des principales composantes à retenir dans le modèle. L'approche des moindres carrés partiels (PLS) demeure une alternative intéressante à MFPCA. Car les composants PLS obtenus sont basés sur la relation entre les prédicteurs (X) et la réponse (Y). Depuis l'introduction de la régression PLS sur des données fonctionnelles univariées dans Preda and Saporta (2002), de nombreuses contributions ont été proposées, principalement dans le cadre fonctionnel univarié. Ainsi, le travail de Preda et al. (2007) a démontré la possibilité d'utiliser la régression PLS pour l'analyse discriminante linéaire. Par-ailleurs, dans Aguilera et al. (2010), les auteurs montrent la relation qui existe entre la procédure PLS

sur des données fonctionnelles univariées et la procédure PLS classique. Celle-ci est faite sur les coefficients obtenus à partir de l'approximation d'expansion de base sur les données explicatives. Une variante des moindres carrés partiels fonctionnels (non itérative) pour la régression a aussi été développée dans Delaigle and Hall (2012b). Cette reformulation permet aux auteurs de faire une étude asymptotique détaillée de la régression PLS. À des fins d'interprétabilité, les auteurs dans Guan et al. (2022) ont récemment introduit une approche PLS modifiée pour favoriser l'obtention d'une fonction coefficient parcimonieuse.
À notre connaissance, la régression PLS pour les données fonctionnelles multivariées n'a été explorée que dans le cadre d'un seul domaine. Dans Dembowska et al. (2021), les auteurs ont proposé une approche en deux étapes pour traiter les covariables fonctionnelles multivariées. La première étape consiste à calculer indépendamment les composantes PLS pour chaque dimension (univariée) $X^{(j)}, j = 1, \ldots, p$. Ensuite, ils en extraient de nouvelles caractéristiques non corrélées basées sur des combinaisons linéaires des composantes PLS obtenues. Dans Beyaztas and Shang (2022), le but est de fournir une version robuste de PLS pour les données fonctionnelles multivariées. Les résultats d'expansion de base de Aguilera et al. (2010) sont étendus sur les données fonctionnelles multivariées. De plus, les auteurs proposent une régression M robuste partielle dans ce cadre.

Notre travail se propose d'étudier plus exhaustivement la procédure PLS, élargissant la récente contribution de Beyaztas and Shang (2022). En effet, nous montrons qu'il existe une relation entre la régression PLS avec des données fonctionnelles univariées (FPLS) et la régression PLS sur des données fonctionnelles multivariées (MFPLS). Cette relation permet d'estimer les composantes PLS pour les données fonctionnelles multivariées à partir des données univariées correspondantes. Dans le cadre d'un seul domaine, elle fournit, d'un point de vue computationnel, une nouvelle procédure d'estimation. Dans le cas de domaines différents, cette relation permet de traiter séparément chaque donnée fonctionnelle univariée avec un domaine différent, puis de combiner les composantes FPLS pour obtenir les composantes MFPLS. Par ailleurs, le fait d'autoriser les covariables fonctionnelles à avoir des domaines de définitions différents permet d'inclure des données fonctionnelles de plusieurs types dans l'analyse supervisée.

La suite du chapitre est organisée en plusieurs sections. La Section 2.2 présente la méthodologie PLS pour la classification binaire et la régression. Elle introduit la régression PLS avec des données fonctionnelles multivariées définies sur différents domaines et établit la relation avec l'approche PLS fonctionnelle univariée. La Section 2.3 présente des études de simulation sur la régression et la classification. Elle permet la comparaison des performances de l'approche proposée MFPLS avec quelques méthodes existantes. La méthode MFPLS est testée sur des données en libre accès pour la classification des séries chronologiques multivariées dans la Section 2.4. Le chapitre se conclut dans la Section 2.5 sur un bilan de la méthode présenté et des perspectives de recherches pour des travaux futurs. L'annexe contient les démonstrations détaillées des résultats théoriques.

2.2 Méthode

2.2.1 Concepts de base et notations

Le cadre des données fonctionnelles multivariées que l'on considère dans cette partie s'inspire du travail Happ and Greven (2018). Comme modèle général pour l'analyse de données fonctionnelles multivariées, considérons X en tant qu'un processus stochastique représenté par un vecteur à p-dimension de variables aléatoires fonctionnelles $X = (X^{(1)}, \ldots, X^{(p)})^\top$ défini sur l'espace de probabilité $(\Omega, \mathcal{A}, \mathbb{P})$.

Habituellement, les composantes $X^{(j)}$, $j = 1, \ldots, p$, sont supposées être des processus stochastiques à valeurs dans $\mathbb{R}$ et définies sur un intervalle continu fini $[0, T]$ (Ramsey and Silverman (2005), Jacques and Preda (2014), Górecki et al. (2015)). Cependant, ici, le cadre considéré suppose que chaque composante $X^{(j)}$ peut être définie sur un domaine spécifique $\mathcal{T}_j$ de $\mathbb{R}^{d_j}$, avec $d_j \in \mathbb{N} - \{0\}$. Ainsi, lorsque les réalisations de $X^{(j)}$ sont des fonctions univariées alors $d_j = 1$ (des séries temporelles par exemple) et $d_j \geq 2$ lorsqu'elles sont des surfaces ou des formes complexes ; en particulier $d_j = 2$ pour des images. On suppose également que $X^{(j)}$ est un processus L_2-continu, c'est-à-dire que chaque trajectoire de $X^{(j)}$ appartient à l'espace de Hilbert des fonctions de carré intégrables définies sur $\mathcal{T}_j$: $L_2(\mathcal{T}_j)$. Ces hypothèses générales garantissent que les intégrales impliquant les variables $X^{(j)}$ soient bien définies.
De cette façon, X prend ses valeurs dans $\mathcal{H}$, où $\mathcal{H} = L_2(\mathcal{T}_1) \times \ldots \times L_2(\mathcal{T}_p)$ est l'espace hilbertien fonctionnel suivant :

$$\mathcal{H} = \{f = (f^{(1)}, \ldots, f^{(p)})^\top, f^{(j)} \in L_2(\mathcal{T}_j)\ , j = 1, \ldots, p\}.$$

Le produit scalaire qui équipe $\mathcal{H}$, noté $\langle , \rangle_{\mathcal{H}}$, est

$$\langle f, g \rangle_{\mathcal{H}} = \sum_{j=1}^{p} \langle f^{(j)}, g^{(j)} \rangle_{L_2(\mathcal{T}_j)} = \sum_{j=1}^{p} \int_{\mathcal{T}_j} f^{(j)}(t) g^{(j)}(t) dt.$$

où dt est la mesure de Lebesgue définie sur $\mathcal{T}_j$. La norme induite par $\langle , \rangle_{\mathcal{H}}$ est notée $\| , \|_{\mathcal{H}}$.

2.2.2 Le modèle de régression linéaire fonctionnel

Lorsque l'objectif est la prédiction (le contexte supervisé), la variable fonctionnelle X est associée à une variable d'intérêt Y par l'espérance conditionnelle $\mathbb{E}(Y|X)$.
Dans un premier temps, considérons que la variable d'intérêt prend ses valeurs dans $\mathbb{R}$. De plus, supposons qu'elle soit de variance finie et définie sur le même espace de probabilité que X

$$Y : \Omega \to \mathbb{R}.$$

Sans perte de généralité, nous supposerons que Y et X sont de moyenne nulle,

$$\mathbb{E}(Y) = 0, \quad \mathbb{E}(X^{(j)}(t)) = 0 \tag{2.1}$$

où $t \in \mathcal{T}_j$ et $j = 1, \ldots, p$.

Le modèle de régression linéaire fonctionnel suppose que $\mathbb{E}(Y|X)$ existe et est un opérateur linéaire en fonction de X. Ainsi, le modèle suivant est considéré :

$$Y = \langle X, \beta \rangle_{\mathcal{H}} + \epsilon, \tag{2.2}$$

où

- $\beta \in \mathcal{H}$ désigne la fonction coefficient (ou paramètre) de régression,

$$\beta = \left(\beta^{(1)}, \ldots, \beta^{(p)}\right)^\top,$$

- ϵ est le terme résiduel supposé de variance finie $\mathbb{E}(\epsilon^2) = \sigma^2 < \infty$ et non corrélé à X.

Le modèle (2.2) peut aussi s'écrire sous sa forme intégrale :

$$Y = \sum_{j=1}^{p} \int_{\mathcal{T}_j} X^{(j)}(t)\beta^{(j)}(t)dt + \epsilon. \tag{2.3}$$

Sous le critère des moindres carrés, l'estimation de la fonction coefficient β est généralement un problème inverse mal posé (Aguilera et al. (2010), Preda and Saporta (2002), Preda et al. (2007)). D'un point de vue théorique, cela est dû à la dimension infinie de X rendant l'opérateur de covariance non inversible (Cardot et al., 1999). Par conséquent, des méthodes de réduction de dimensions telles que l'analyse en composantes principales (Happ and Greven (2018)) et l'expansion de X dans une base de fonctions (Aguilera et al. (2010)) peuvent être utilisées afin d'obtenir une approximation de β dans (2.3).

Expansion (de X) dans une base de fonctions

Pour $j = 1, \ldots, p$, considérons un ensemble $\Psi^{(j)} = \{\psi_1^{(j)}, \ldots, \psi_{M_j}^{(j)}\}$ de M_j fonctions linéairement indépendantes dans $L_2(\mathcal{T}_j)$. De plus, supposons que X et β admettent les écritures suivantes :

$$X^{(j)}(t) = \sum_{k=1}^{M_j} a_k^{(j)} \psi_k^{(j)}(t), \quad \beta^{(j)}(t) = \sum_{k=1}^{M_j} b_k^{(j)} \psi_k^{(j)}(t), \tag{2.4}$$

où $t \in \mathcal{T}_j$ et $j = 1, \ldots, p$.
Ainsi l'équation (2.3) est équivalente au modèle de régression linéaire multiple :

$$Y = (\mathrm{F}a)^\top b + \epsilon \tag{2.5}$$

où $M = \sum_{j=1}^{p} M_j$, a et b sont les vecteurs de taille M suivants :

$$a = \begin{pmatrix} a^{(1)} \\ \ldots \\ a^{(p)} \end{pmatrix} \text{ et } b = \begin{pmatrix} b^{(1)} \\ \ldots \\ b^{(p)} \end{pmatrix}.$$

où $a^{(j)} = (a_1^{(j)}, a_2^{(j)}, \ldots, a_{M_j}^{(j)})^\top$ et $b^{(j)} = (b_1^{(j)}, b_2^{(j)}, \ldots, b_{M_j}^{(j)})^\top$.
La matrice F de taille $M \times M$ est composée de p-blocs diagonaux

$$\mathrm{F} = \begin{pmatrix} \mathrm{F}^{(1)} & 0 & \ldots & 0 \\ 0 & \mathrm{F}^{(2)} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \mathrm{F}^{(p)} \end{pmatrix}$$

où

$$\mathrm{F}^{(j)} = \begin{pmatrix} \langle \psi_1^{(j)}, \psi_1^{(j)} \rangle_{\mathcal{L}(\mathcal{T}_j)} & \ldots & \langle \psi_1^{(j)}, \psi_l^{(j)} \rangle_{\mathcal{L}(\mathcal{T}_j)} & \ldots & \langle \psi_1^{(j)}, \psi_p^{(j)} \rangle_{\mathcal{L}(\mathcal{T}_j)} \\ \vdots & & \ldots & & \vdots \\ \langle \psi_k^{(j)}, \psi_1^{(j)} \rangle_{\mathcal{L}(\mathcal{T}_j)} & \ldots & \langle \psi_k^{(j)}, \psi_l^{(j)} \rangle_{\mathcal{L}(\mathcal{T}_j)} & \ldots & \langle \psi_k^{(j)}, \psi_p^{(j)} \rangle_{\mathcal{L}(\mathcal{T}_j)} \\ \vdots & & \ldots & & \vdots \\ \langle \psi_p^{(j)}, \psi_1^{(j)} \rangle_{\mathcal{L}(\mathcal{T}_j)} & \ldots & \langle \psi_p^{(j)}, \psi_l^{(j)} \rangle_{\mathcal{L}(\mathcal{T}_j)} & \ldots & \langle \psi_p^{(j)}, \psi_p^{(j)} \rangle_{\mathcal{L}(\mathcal{T}_j)} \end{pmatrix}.$$

Par conséquent, sous l'hypothèse (2.4), l'estimation de β est équivalente à celle du vecteur b dans un modèle de régression linéaire multiple. La matrice des variables explicatives est composée des coefficients d'expansion de base de X (le vecteur a) et de la métrique fournie par le choix des fonctions de la base (la matrice F).
Cependant, estimer b sous le critère des moindres carrés peut, dans certains contextes (par exemple, un grand nombre de fonctions de base), conduire à des problèmes de multicolinéarité et de grande dimension, comme dans le cas univarié (Aguilera et al., 2010). Deux méthodes d'estimation : la régression en composantes principales (PCR) et la régression des moindres carrés partiels (PLS), sont réputées pour l'efficacité de leur algorithme d'estimation et l'interprétabilité des résultats. Comme mentionné dans Jong (1993) pour le cadre multivarié classique et dans Aguilera et al. (2010) pour le cas univarié fonctionnel, pour un nombre fixe de composantes, la régression PLS donne des meilleurs résultats que la PCR. En effet, la régression PLS fournit une solution plus efficace en termes de critère de la somme des erreurs au carré. Des études numériques confirment ces résultats pour la régression sur des données fonctionnelles univariées (Delaigle and Hall (2012b), Guan et al. (2022)).
Dans la section suivante, nous présentons la méthode de régression PLS sur les données fonctionnelles multivariées.

2.2.3 La régression sur les données fonctionnelles multivariées : MF-PLS

La régression PLS pénalise le critère des moindres carrés en maximisant la covariance entre des combinaisons linéaires, obtenues à l'aide des fonctions poids w, de la variables explicative X et de la variable à expliquer Y. Elle est basée sur un algorithme itératif construisant à chaque étape des composantes dites PLS servant de variables explicatives au modèle final. Lorsque X est multivarié, comme dans le cas univarié (Preda and Saporta (2002)), les fonctions w sont obtenues comme solutions du critère de Tucker :

$$\max_{w \in \mathcal{H}} \mathrm{Cov}^2(\langle w, X \rangle_{\mathcal{H}}, Y), \tag{2.6}$$

avec $w = (w^{(1)}, \ldots, w^{(p)})^\top$ tel que $||w||_{\mathcal{H}} = 1$.

La proposition qui suit établit la solution au problème de maximisation ci-dessus.

Proposition 1. *La solution de* (2.6) *est donnée par*

$$w^{(j)}(t) = \frac{\mathbb{E}(X^{(j)}(t)Y)}{\sqrt{\sum_{k=1}^{p} \int_{\mathcal{I}_k} \mathbb{E}^2(X^{(k)}(s)Y)ds}}, \forall t \in \mathcal{T}_j,\ j = 1, \ldots, p. \tag{2.7}$$

La première composante PLS ξ est le résultat du produit scalaire de la fonction w, solution de (2.6), et X :

$$\xi = \langle X, w \rangle_{\mathcal{H}} = \sum_{k=1}^{p} \int_{\mathcal{T}_k} X^{(k)}(t) w^{(k)}(t) dt.$$

L'algorithme itératif PLS fonctionne comme suit :

- Étape 0 : Soit $X_0 = X$ et $Y_0 = Y$.
- Étape h, $h \geq 1$: Définir w_h comme dans la Proposition 1 avec $X = X_{h-1}$ et $Y = Y_{h-1}$. Ensuite, calculer la h-ième composante PLS comme :

  $$\xi_h = \langle X_{h-1}, w_h \rangle_{\mathcal{H}},$$

  Déterminer les résidus X_h et Y_h de la régression linéaire de X_{h-1} et Y_{h-1} sur ξ_h,

  $$\begin{aligned} X_h &= X_{h-1} - \rho_h \xi_h, \\ Y_h &= Y_{h-1} - c_h \xi_h, \end{aligned}$$

  où $\rho_h = \dfrac{\mathbb{E}(X_{h-1}\xi_h)}{\mathbb{E}(\xi_h^2)} \in \mathcal{H}$ et $c_h = \dfrac{\mathbb{E}(Y_{h-1}\xi_h)}{\mathbb{E}(\xi_h^2)} \in \mathbb{R}$.
- Passer à l'étape suivante ($h = h + 1$).

De plus, les propriétés suivantes, énoncées dans le cadre univarié (Proposition 3 dans Preda and Saporta (2002)), sont toujours valables.

Proposition 2. *Pour tout $h \geq 1$, $\{\xi_k\}_{k=1}^{h}$ forme un système orthogonal de $L(X)$ et implique les formules d'expansion suivantes :*

$$Y = c_1\xi_1 + c_2\xi_2 + \ldots + c_h\xi_h + Y_h,$$

$$X^{(j)}(t) = \rho_1^{(j)}(t)\xi_1 + \rho_2^{(j)}(t)\xi_2 + \ldots + \rho_h^{(j)}(t)\xi_h + X_h^{(j)}(t),$$
$$t \in \mathcal{T}_j,\ j = 1, \ldots, p\,,$$

où $L(X)$ désigne l'espace engendré par X.

Le terme de droite du développement de Y fournit l'approximation PLS d'ordre h de (2.2),

$$\langle X, \beta \rangle_{\mathcal{H}} \approx c_1\xi_1 + c_2\xi_2 + \ldots + c_h\xi_h \tag{2.8}$$

et la partie résiduelle :

$$\epsilon \approx Y_h.$$

Néanmoins, les propriétés ci-dessus ne donnent pas une relation claire entre Y et X comme dans (2.2). Cela est dû au fait qu'après le premier pas de l'algorithme, ξ_h est calculé à partir de X_h et non plus de X. Pour obtenir une telle relation, il nous faut écrire ξ_h en fonction de X.

Lemme 1. Soit $\{v_k\}_{k=1}^h$, $v_k \in \text{span}\{w_1, \ldots, w_h\}$:

$$v_h^{(j)}(t) = w_h^{(j)}(t) - \sum_{k=1}^{h-1} \langle \rho_k, w_h \rangle_{\mathcal{H}} v_k^{(j)}(t),\ t \in \mathcal{T}_j\ j = 1, \ldots, p. \tag{2.9}$$

Alors, $\{v_k\}_{k=1}^h$ forme un système de fonctions linéairement indépendantes dans $\mathcal{H}$ et

$$\xi_h = \langle v_h, X \rangle_{\mathcal{H}} = \sum_{j=1}^{p} \int_{\mathcal{T}_j} X^{(j)}(t) v_h^{(j)}(t) dt,$$

où span$\{w_1, \ldots, w_h\}$ *désigne l'espace engendré par* $w_1, \ldots, w_h$.

Ainsi, comme pour l'analyse en composantes principales de X (Happ and Greven, 2018), la régression PLS calcule les composantes en tant que produit scalaire de X et des fonctions $\{v_k\}_{k=1}^h$. Ces composantes conviennent à la régression, car capturant le maximum d'informations entre X et Y, selon le critère de Tucker. Il revient alors d'après le Lemme 1 et (2.8) que :

$$c_1\xi_1 + c_2\xi_2 + \ldots + c_h\xi_h = \langle X, \beta_h \rangle_{\mathcal{H}}$$

avec :

$$\beta_h = \sum_{i=1}^{h} c_i v_i. \tag{2.10}$$

Ainsi, le modèle de régression PLS obtenu avec les h premières composantes est donné par :

$$Y = \langle X, \beta_h \rangle_{\mathcal{H}} + \varepsilon_h,$$

avec $\varepsilon_h = Y_h$.

Remarque 1. *Pour* $h \geq 1$, *soient* $\mathcal{V}_h = \begin{pmatrix} v_1 & \ldots & v_h \end{pmatrix}$ *et* $\mathcal{W}_h = \begin{pmatrix} w_1 & \ldots & w_h \end{pmatrix}$ *deux vecteurs (lignes) de* $\mathcal{H}^h$. *À partir du Lemme 1, on déduit la relation entre* $\mathcal{V}_h$ *et* $\mathcal{W}_h$

$$\mathcal{V}_h = \mathcal{W}_h - \mathrm{P}_h \mathcal{V}_h, \tag{2.11}$$

où P_h *est la matrice* $h \times h$,

$$\mathrm{P}_h = \begin{pmatrix} 0 & \langle \rho_1, w_2 \rangle_{\mathcal{H}} & \langle \rho_1, w_3 \rangle_{\mathcal{H}} & \ldots & \langle \rho_1, w_{h-1} \rangle_{\mathcal{H}} & \langle \rho_1, w_h \rangle_{\mathcal{H}} \\ 0 & 0 & \langle \rho_2, w_3 \rangle_{\mathcal{H}} & \ldots & \langle \rho_2, w_{h-1} \rangle_{\mathcal{H}} & \langle \rho_2, w_h \rangle_{\mathcal{H}} \\ \vdots & \vdots & \ddots & \ldots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & 0 & \langle \rho_{h-1}, w_h \rangle_{\mathcal{H}} \\ 0 & 0 & 0 & \ldots & 0 & 0 \end{pmatrix}.$$

Puisque $\mathbb{I}_{h\times h} + \mathrm{P}_h$ *est inversible, l'équation* (2.11) *est équivalente à :*

$$\mathcal{V}_h = (\mathbb{I}_{h\times h} + \mathrm{P}_h)^{-1}\mathcal{W}_h, \tag{2.12}$$

où $\mathbb{I}_{h\times h}$ *est la matrice identité de taille* $h \times h$*. Comme les fonctions* $\{w_k\}_{k=1}^h$ *sont calculées directement à partir de la Proposition 1, l'équation (2.12) fournit un moyen simple d'obtenir les fonctions* $\{v_k\}_{k=1}^h$ *et donc par (2.10) l'approximation de la fonction coefficient* β_h*.*

Remarque 2. *Il convient de noter que, contrairement aux fonctions obtenues par l'ACP (Happ and Greven, 2018),* $\{v_k\}_{k=1}^h$ *n'est pas un système orthogonal par le produit scalaire* $\langle , \rangle_{\mathcal{H}}$*. Néanmoins, il fournit des composantes PLS orthogonales, c'est-à-dire* $\mathbb{E}(\xi_k\xi_l) = \mathbb{E}(\xi_k^2)\delta_{k,l}$ *, où* $\delta_{k,l}$ *est le symbole de Kronecker.*

La partie suivante se concentre sur la relation entre la régression des moindres carrés partiels des données fonctionnelles univariées (FPLS) et la version multivariée proposée (MFPLS). Plus précisément, nous montrons que la régression MFPLS peut être résolue en itérant FPLS dans une approche en deux étapes.

Relation entre MFPLS et FPLS

Pour $j = 1, \dots p$, soit $\tilde{w}_1^{(j)}$ la fonction poids correspondant à la première composante PLS obtenue par régression FPLS de Y sur la dimension $X^{(j)}$ (Proposition 2 dans Preda and Saporta (2002)),

$$\tilde{w}_1^{(j)}(t) = \frac{\mathbb{E}(X^{(j)}(t)Y)}{\sqrt{\int_{\mathcal{T}_j}\mathbb{E}^2(X^{(j)}(s)Y)ds}},\ t \in \mathcal{T}_j.$$

On peut remarquer que les fonctions $\tilde{w}_1^{(j)}$ et $w_1^{(j)}$ (telle que définie dans la Proposition 1) vérifient la relation suivante :

$$w_1^{(j)} = u_{1,j}\tilde{w}_1^{(j)},$$

avec $u_{1,j} = \dfrac{||\mathbb{E}(X^{(j)}Y)||_{L_2(\mathcal{T}_j)}}{|||\mathbb{E}(XY)|||} \in \mathbb{R}$ où $||\cdot||_{L_2(\mathcal{T}_j)}$ représente la norme induite par le produit scalaire (usuel) de $L_2(\mathcal{T}_j)$.

Notons que le vecteur $u_1 = \left(u_{1,1}, ..., u_{1,p}\right)^\top$ est tel que $||u||_{\mathbb{R}^p} = 1$. Par conséquent, nous pouvons établir une relation entre les composantes de MFPLS et FPLS. *Pour* $j = 1, \dots, p$*, notons* $\xi_1^{(j)}$ *la première composante obtenue par la régression FPLS de* Y *sur la* j*- ième dimension de* X*. La première composante MFPLS de* Y *sur* X *:* ξ_1 *s'obtient comme la première composante de la régression PLS de* Y *sur* $\{\xi_1^{(1)}, \dots, \xi_1^{(p)}\}$*.*
Ainsi, comme l'algorithme PLS est itératif, cette relation peut être appliquée au calcul des composants MFPLS d'ordre supérieur.

Sur la base de cette relation, une nouvelle méthodologie fondée sur l'hypothèse (2.4) est présentée dans la partie suivante.

Utilisation de l'expansion en bases de fonctions dans l'algorithme MFPLS

Sous l'hypothèse (2.4), pour $j = 1, \dots p$, notons par $\Lambda^{(j)}$ le vecteur

$$\Lambda^{(j)} = \left(\mathrm{F}^{(j)}\right)^{1/2} a^{(j)}$$

où $\left(\mathrm{F}^{(j)}\right)^{1/2}$ est la matrice racine carrée de $\mathrm{F}^{(j)}$ et $a^{(j)}$ est le vecteur de coefficients de l'approximation de $X^{(j)}$ dans la base des fonctions $\{\psi_1^{(j)}, \dots, \psi_{M_j}^{(j)}\}$. Alors, la Proposition 2 dans Aguilera et al. (2010) rend la procédure MFPLS équivalente à l'algorithme suivant.

L'algorithme MFPLS

- Étape 0 : Soit $\Lambda_0^{(j)} = \Lambda^{(j)}$ pour $j = 1, \dots, p$ et $Y_0 = Y$.
- Étape h, $h \geq 1$:
  1. Pour $j = 1, \dots, p$,
     - Définir $\xi_h^{(j)}$ comme la **première** composante PLS obtenue par la régression ordinaire PLS de Y_{h-1} sur $\Lambda_{h-1}^{(j)}$,

$$\xi_h^{(j)} = \sum_{k=1}^{M_j} \Lambda_{h-1,k}^{(j)} \theta_{h,k}^{(j)}, \tag{2.13}$$

     où $\theta_h^{(j)} \in \mathbb{R}^{M_j}$ est le vecteur poids associé.
  2. Définir ξ_h, la h-ième composante MFPLS, comme la première composante de la régression de Y_h sur $\{\xi_h^{(1)}, \dots, \xi_h^{(p)}\}$,

$$\xi_h = \sum_{k=1}^{p} \xi_h^{(k)} u_{h,k}, \tag{2.14}$$

     où $u_h \in \mathbb{R}^p$ est le vecteur poids associé.
  3. – Pour $j = 1, \dots, p$, calculer les résidus $\Lambda_h^{(j)}$ de la régression linéaire de $\Lambda_{h-1}^{(j)}$ sur ξ_h,

$$\Lambda_h^{(j)} = \Lambda_{h-1}^{(j)} - r_h^{(j)} \xi_h,$$

     où $r_h^{(j)} = \dfrac{\mathbb{E}(\xi_h \Lambda_{h-1}^{(j)})}{\mathbb{E}(\xi_h^2)} \in \mathbb{R}^{M_j}$.

     – Calculer le résidu Y_h de la régression linéaire de Y_{h-1} sur ξ_h,

$$Y_h = Y_{h-1} - c_h \xi_h,$$

     où $c_h = \dfrac{\mathbb{E}(Y_{h-1} \xi_h)}{\mathbb{E}(\xi_h^2)} \in \mathbb{R}$.
- Passer à l'étape suivante ($h = h + 1$).

Remarque 3.

1. *Le nombre de composantes PLS (h) retenues dans l'approximation du modèle de régression (2.8) est généralement choisi par validation croisée en optimisant certains critères tels que le MSE*[1] *ou l'AUC*[2] *(pour la classification binaire).*
2. *L'approche de Beyaztas and Shang (2022) est une extension du résultat de Aguilera et al. (2010). Elle a été proposée dans le cadre d'un domaine unique. Notre approche est plus flexible puisqu'elle autorise les dimensions de X à avoir des domaines différents. Le cas d'un seul domaine est donc un cas particulier de la méthodologie proposée (voir la section 2.3.1 pour une comparaison numérique).*
3. *Introduisons à l'étape h - point 3 de l'algorithme le calcul des fonctions $w_h^{(j)}$ et $\rho_h^{(j)}$, $j = 1, \ldots, p$, sous forme de :*

$$w_h^{(j)} = u_{h,j} \mathrm{H}^{(j)} \theta_h^{(j)} \psi^{(j)},$$
$$\rho_h^{(j)} = \mathrm{H}^{(j)} r_h^{(j)} \psi^{(j)}$$

   où $\mathrm{H}^{(j)} = \left(\mathrm{F}^{(j)}\right)^{-1/2}$.
   Ensuite, à l'aide du Lemme 1, les fonctions $\{v_k\}_{k=1}^h$ peuvent également être calculées au moyen de l'algorithme MFPLS, ce qui permet l'obtention de la fonction coefficient, voir (2.10).
4. ***Estimation :*** *Considérons $(\mathcal{X}_1, \mathcal{Y}_1), \ldots, (\mathcal{X}_n, \mathcal{Y}_n)$ un échantillon i.i.d. de taille $n \geq 1$ de (X, Y). Alors, pour $j = 1, \ldots, p$, le vecteur $a^{(j)}$ est représenté par la matrice de l'échantillon $n \times M_j$ de coefficients $\mathbf{A}^{(j)}$*

$$\mathbf{A}^{(j)} = \begin{pmatrix} a_{1,1}^{(j)} & \ldots & a_{1,l}^{(j)} & \ldots & a_{1,M_j}^{(j)} \\ \vdots & & \vdots & & \vdots \\ a_{k,1}^{(j)} & \ldots & a_{k,l}^{(j)} & \ldots & a_{k,M_j}^{(j)} \\ \vdots & & \vdots & & \vdots \\ a_{n,1}^{(j)} & \ldots & a_{n,l}^{(j)} & \ldots & a_{n,M_j}^{(j)}. \end{pmatrix}$$

   et

$$\mathbf{\Lambda}^{(j)} = \mathbf{A}^{(j)} (\mathrm{F}^{(j)})^{1/2}.$$

   Définissons $\mathbf{Y} = (\mathcal{Y}_1, \ldots, \mathcal{Y}_n)^\top$, la version matricielle de l'algorithme MFPLS (étape h) peut alors être réécrite comme suit.

   1 *Pour chaque $j = 1, \ldots, p$, définir $\boldsymbol{\xi}_h^{(j)} \in \mathbb{R}^n$ comme la première composante PLS obtenue par la PLS ordinaire de $\mathbf{Y}_h$ sur $\mathbf{\Lambda}_{h-1}^{(j)}$.*
   2 *La h-ième composante $\boldsymbol{\xi}_h$ est la première composante obtenue par la PLS ordinaire de $\mathbf{Y}_h$ sur $\left(\boldsymbol{\xi}_h^{(1)}, \ldots, \boldsymbol{\xi}_h^{(p)}\right)$.*
      Pour $j = 1, \ldots, p$, les résidus $\mathbf{\Lambda}_h^{(j)}$ sont calculés par

$$\mathbf{\Lambda}_h^{(j)} = \mathbf{\Lambda}_{h-1}^{(j)} - \boldsymbol{\xi}_h \mathbf{r}_h^{(j)}$$

      avec $\mathbf{r}_h^{(j)} = \dfrac{1}{\boldsymbol{\xi}_h^\top \boldsymbol{\xi}_h} \boldsymbol{\xi}_h^\top \mathbf{\Lambda}_{h-1}^{(j)}$ est le coefficient de projection.
      Et le résidu $\mathbf{Y}_h$ est :

$$\mathbf{Y}_h = \mathbf{Y}_{h-1} - \boldsymbol{\xi}_h \mathbf{c}_h$$

1. Mean Square Error
2. Area under the curve

$$avec\ \mathbf{c}_h = \frac{1}{\boldsymbol{\xi}_h^\top \boldsymbol{\xi}_h} \boldsymbol{\xi}_h^\top \mathbf{Y}_{h-1}.$$

Bien que la méthodologie présentée soit originalement proposée pour la prédiction d'une variable réelle, elle peut être utilisée pour la classification binaire en exploitant la relation entre l'analyse discriminante linéaire et la régression linéaire (Aguilera et al. (2010), Preda et al. (2007)). La section suivante traite d'une approche de classification basée sur la régression PLS.

De la régression PLS à la classification binaire

Dans cette partie, la variable à prédire Y est binaire. La classification supervisée suppose que Y est une variable de Bernoulli : $Y \in \{0, 1\}$ et $Y \sim \mathcal{B}(\pi_1)$ où $\pi_1 = \mathbb{P}(Y = 1)$. La régression PLS peut être étendue à la classification binaire à l'aide d'un recodage de Y. Soit Y^* défini comme :

$$Y^* = \begin{cases} \sqrt{\frac{\pi_1}{\pi_0}} & \text{, si } Y = 0 \\ -\sqrt{\frac{\pi_0}{\pi_1}} & \text{sinon} \end{cases} \tag{2.15}$$

où $\pi_0 = 1 - \pi_1$. Alors, la fonction coefficient β de la régression de Y^* sur X (pas nécessairement centré) correspond à une constante près à celle définissant le score discriminant de Fisher désigné par $\Gamma(X)$:

$$\Gamma(X) = \alpha + \langle X, \beta \rangle_{\mathcal{H}},$$

avec α= -$\langle \mu, \beta \rangle_{\mathcal{H}}$ et $\mu = \mathbb{E}(X) \in \mathcal{H}$.
Finalement, la classe prédite $\hat{Y}_0$ d'une nouvelle courbe X_0 est donnée par

$$\hat{Y}_0 = \begin{cases} 0 & \text{si } \Gamma(X_0) > 0 \\ 1 & \text{sinon.} \end{cases}$$

Cette procédure est expliquée en détails dans Preda et al. (2007). Dans ce chapitre, nous estimons la fonction coefficient β par l'approche MFPLS.

2.3 Études de simulation

Cette section présente quelques études numériques de la méthode MFPLS à l'aide de données simulées. Ses performances sont comparées à celles obtenues par les méthodes basées sur l'analyse en composante principale (MFPCA). Deux cas sont étudiés : (i) toutes les composantes $X^{(j)}$ de X sont définies sur le même domaine $\mathcal{T} = [0, T]$, $T > 0$ et (ii) X est un vecteur fonctionnel bivarié $X = (X^{(1)}, X^{(2)})^\top$ avec $X^{(1)} = (X^{(1)}(t))_{t \in \mathcal{T}_1}$ et $X^{(2)} = (X^{(2)}(t))_{t \in \mathcal{T}_2 \times \mathcal{T}_2}$, où $\mathcal{T}_1, \mathcal{T}_2 \subset \mathbb{R}$. Ainsi, dans (ii) un échantillon de la variable fonctionnelle X correspond à un ensemble de courbes et d'images.

2.3.1 Cas d'un unique domaine : régression

Étant donné que la principale méthode concurrente de MFPLS a été proposée pour la régression (Beyaztas and Shang, 2022), nous appliquons dans cette section le cadre de simulation de régression décrit dans Beyaztas and Shang (2022) et comparons leur méthode avec MFPLS.
Soient le domaine $\mathcal{T} = [0, 1]$ et $X = (X^{(1)}, X^{(2)}, X^{(3)})^\top$, où

$$X^{(j)}(t) = \sum_{k=1}^{5} \gamma_k \upsilon_k(t), \quad t \in \mathcal{T}, \ j = 1, 2, 3,$$

$\gamma_k \sim \mathcal{N}(0, 4k^{-3/2})$ et $\upsilon_k(t) = \sin k\pi t - \cos k\pi t$, $k = 1, \ldots, 5$.
La fonction β est définie comme :

$$\beta(t) = \big(\sin(2\pi t), \sin(3\pi t), \cos(2\pi t)\big)^\top.$$

Le modèle de régression générant les données est :

$$Y = \langle X, \beta \rangle_{\mathcal{H}} + \epsilon,$$

où $\epsilon \sim \mathcal{N}(0, \sigma^2)$.
Nous définissons la variance du bruit comme :

$$\sigma^2 = \frac{\mathbb{E}(\langle X, \beta \rangle_{\mathcal{H}}^2)}{\text{SNR}},$$

où SNR est le rapport signal sur bruit. Les cinq valeurs suivantes de SNR sont considérées : 0.5, 1.62, 2.75, 3.88 et 5.
L'approche proposée dans Beyaztas and Shang (2022) est une généralisation du résultat de Aguilera et al. (2010) au cas multivarié (MFPLS_D). Elle exploite une équivalence entre la régression PLS sur des covariables fonctionnelles multivariées et la régression PLS ordinaire sur les scores de projection des covariables dans les bases de fonctions. Notre méthode a une procédure différente. Elle calcule les composantes PLS multivariées à l'aide des composantes PLS univariées (voir la section 2.2.3).

Comme dans Beyaztas and Shang (2022), on considère 200 points de temps discrets équidistants sur $\mathcal{T}$ où les données brutes de X sont observées, et 400 réalisations indépendantes du couple (X, Y) sont simulées. Parmi elles, 50% sont utilisées pour l'apprentissage et celles qui restent composent l'échantillon test. En plus de l'approche MFPLS_D, notre modèle (MFPLS) est également comparé à la régression en composantes principales (MFPCR) [3].
L'erreur quadratique moyenne de prédiction (MSPE) sert à mesurer les performances des trois approches sur les données tests.

$$\text{MSPE} = \frac{1}{200} \sum_{i \in V_{\text{set}}} (Y_i - \hat{Y}_i)^2,$$

3. Implémentation : https://github.com/UfukBeyaztas/RFPLS

où $\hat{Y}_i$ est la prédiction de Y_i, la i-ème observation de Y dans V_{set} l'échantillon test.
Les procédures d'estimations des trois méthodes sont réalisées sur 200 réplications de l'expérience. Le nombre de composantes pour toutes les approches est choisi par des validations croisées à 10-folds. Pour transformer les données brutes en fonctions, une base composée de 20 fonctions splines quadratiques est utilisée.

Le tableau 2.1 présente un résumé des résultats (MSPE) obtenus par les trois approches MFPLS, MFPLS_D et MFPCR.

SNR	MFPLS	MFPLS_D	MFPCR
	MSPE	**MSPE**	**MSPE**
0.50	10.85(1.56)	10.63(1.48)	10.73(1.48)
1.62	3.42(0.45)	3.43(0.45)	3.45(0.45)
2.75	2.04(0.30)	2.07(0.29)	2.08(0.31)
3.88	1.45(0.26)	1.47(0.22)	1.47(0.22)
5.00	1.11(0.16)	1.15(0.17)	1.15(0.18)

Table 2.1 – Moyennes et écarts types (entre parenthèses) des MSPE obtenus lors des expériences.

Les méthodes testées donnent des résultats comparables. Ainsi, notre approche et l'approche directe donnent des MSPE proches.

2.3.2 Cas de domaines différents : classification

La méthode proposée dans ce chapitre permet de prendre en compte simultanément des images et des séries temporelles dans un but supervisé. Cette partie se concentre sur l'étude d'un cas de classification avec des données de domaines différents.

Soient les intervalles $\mathcal{T}_1 = [0, 50]$, $\mathcal{T}_2 = [0, 1] \times [0, 1]$, et $X = (X^{(1)}, X^{(2)})^\top$, où

$$X^{(1)}(t) = Z_1 h(t) + \epsilon^{(1)}(t), t \in \mathcal{T}_1$$
$$X^{(2)}(t) = Z_2 q(t) + \epsilon^{(2)}(t), t \in \mathcal{T}_2.$$

La variable fonctionnelle $\epsilon = (\epsilon^{(1)}, \epsilon^{(2)})^\top$ est composée de deux dimensions indépendantes. La première ($\epsilon^{(1)}$) est un bruit blanc de variance σ^2, tandis que la seconde $\epsilon^{(2)}$ est un champ aléatoire gaussien. Cette dernière est associée à un modèle de covariance de Matern, avec des paramètres de seuil, de portée et de pépite respectivement égaux à 0.25, 0.75 et σ, voir Ribeiro Jr et al. (2001) pour plus de détails. Les termes Z_1 et Z_2 sont des variables de Bernoulli, avec des valeurs dans $\{0, 1\}$. Les fonctions (déterministes) h et q sont données par :

$$h(t) = 3.14 \left(1 - \frac{|t - 10|}{4}\right)_+ \qquad q(s) = -2 \log \left(\sqrt{(s^{(1)} - 0.5)^2 + (s^{(2)} - 0.5)^2}\right)_+$$

où $(.)_+$ désigne la partie positive, $t \in \mathcal{T}_1$ et $s = (s^{(1)}, s^{(2)}) \in \mathcal{T}_2$.
La variable réponse Y est construite comme suit :

$$Y = \begin{cases} 1 & \text{si } Z_1 Z_2 = 1 \\ 0 & \text{sinon.} \end{cases}$$

En d'autres termes, $Y = 1$ lorsque Z_1 et Z_2 sont toutes deux égales à 1 comme illustré dans la Figure 2.1.

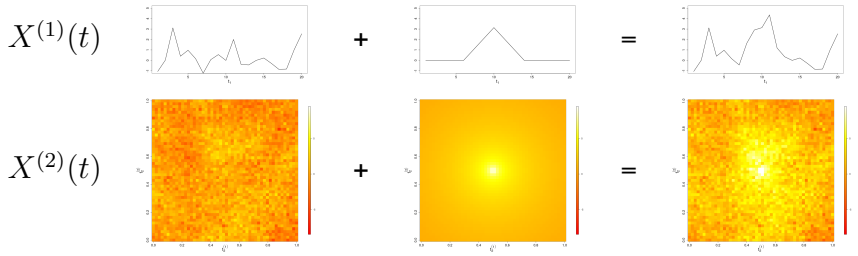

Figure 2.1 – Construction de la classe $Y = 1$ avec SNR= 0.5.

Pour la première et la deuxième dimension, les grilles d'observations sont respectivement 50 points équidistants et 50×50 pixels. Afin d'obtenir la forme fonctionnelle de X, les dimensions sont projetées dans des bases composées de 20 fonctions splines quadratiques pour la première et de 4 fonctions splines bidimensionnelles pour la seconde (Happ, 2017). Les variances des fonctions q et h tout le long de leur domaine sont approximativement de 1. Le rapport signal/bruit (SNR) est donc (presque) le même sur les deux dimensions et ne dépend que de σ

$$\text{SNR} = \frac{1}{\sigma^2}.$$

En agissant sur le paramètre σ, nous considérons plusieurs valeurs de SNR : 0.5, 0.7, 1.2, 2.1 et 4.9.

Les probabilités $\mathbb{P}(Z_1 = 1)$ et $\mathbb{P}(Z_2 = 1)$ sont fixées à $3/4$, de façon à avoir la répartition suivante des classes : $\mathbb{P}(Y = 1) = 9/16 \simeq 0.56$. Nous simulons 500 réalisations indépendantes du couple (X, Y). Pour l'apprentissage, nous utilisons les 75% et les 25% restantes sont destinées à l'ensemble de test.

Pour chaque valeur de SNR, trois modèles MFPLS sont calculés. Les deux premiers prennent en compte exclusivement une dimension de la variable explicative. Ils sont désignés par MFPLS(1) et MFPLS(2), et utilisent respectivement les variables $X^{(1)}$ et $X^{(2)}$. Le troisième, désigné simplement par MFPLS, se sert des deux dimensions simultanément pour la classification.
L'objectif est d'évaluer les performances des approches de domaine unique par rapport à celle de domaines multiples. L'approche MFPCA-LDA sert de modèle concurrent aux approches MFPLS. Elle fait l'analyse linéaire discriminante à l'aide des scores obtenus par l'analyse en composantes principales Happ (2017).
Le nombre de composantes dans les approches MFPLS et MFPCA-LDA est choisi par une validation croisée à 10-folds en utilisant l'AUC. Les modèles sont évalués par l'AUC obtenu sur l'ensemble test. L'expérience est répétée 200 fois.

SNR	MFPLS	MFPLS(1)	MFPLS(2)	MFPCA-LDA
0.50	0.93(0.03)	0.73(0.04)	0.80(0.04)	0.88(0.05)
0.73	0.95(0.02)	0.75(0.05)	0.81(0.04)	0.95(0.02)
1.16	0.97(0.01)	0.77(0.04)	0.81(0.04)	0.97(0.02)
2.10	0.98(0.01)	0.82(0.04)	0.80(0.04)	0.99(0.01)
4.94	1.00(0.01)	0.85(0.04)	0.81(0.04)	0.99(0.01)

Table 2.2 – Moyennes et écarts-types (en parenthèses) des AUC obtenus durant les 200 experiences.

Le tableau 2.2 montre que MFPLS donne de meilleurs résultats que MFPCA-LDA pour la valeur la plus faible du SNR. Par ailleurs, la différence entre les méthodes disparaît avec l'augmentation du SNR. L'utilisation partielle des données (modèles MFPLS(1) et MFPLS(2)) pour prédire Y est moins efficace que l'utilisation des deux dimensions. Ainsi, le tableau 2.2 prouve clairement l'avantage d'utiliser les deux dimensions de la variable fonctionnelle.

Cette simulation démontre la capacité de notre méthode à classer les données de différents domaines. En outre, comme elle est spécialement conçue pour l'apprentissage supervisé, elle peut être plus efficace que les techniques basées sur l'analyse en composantes principales, telles que MFPCA-LDA, lorsque les données sont bruitées.

2.4 Application : classification de séries temporelles

Dans cette section, nous comparons la classification basée sur la régression PLS avec des méthodes black box (LSTM, Random Forest, etc.) sur des données en libre accès. Elles sont présentées dans la Table 2.3 qui reprend les informations de la Table 1 dans (Karim et al., 2019). Ces données vont de la reconnaissance de caractères à la classification d'enregistrements d'électroencéphalogramme (EEG). Ce sont des séries temporelles fonctionnelles multivariées qui ont été utilisées dans divers travaux pour évaluer de nouvelles méthodologies (voir par exemple Pei et al. (2017), Schäfer and Leser (2017)).

Le modèle proposé (MFPLS) est comparé à des modèles non fonctionnels et à l'analyse discriminante MFPCA-LDA. Cette dernière utilise les scores obtenus par l'analyse en composantes principales fonctionnelles multivariées (Happ (2017)) pour l'analyse discriminante. Les modèles non fonctionnels : le *Long Short-Term Memory Fully Convolutional Network* (LSTM-FCN) et *Attention LSTM-FCN* (ALSTM-FCN) ont été proposés dans Karim et al. (2019). De plus, la colonne SOTA dans le tableau 2.4 présente les compétiteurs de référence de ces derniers modèles. Plus particulièrement, elle donne les meilleures performances obtenues parmi les modèles Dynamic time warping (DTW), Random Forest (RF), SVM avec un noyau linéaire, SVM avec un noyau polynomial de 3^e degré (SVM-Poly), et d'autres méthodes détaillées dans Karim et al. (2019).

L'objectif de cette partie est de montrer que notre modèle est compétitif, bien qu'il soit basé sur la régression.
La répartition des données en échantillons d'apprentissage et de test est la même que celle utilisée dans Karim et al. (2019). Les différents modèles mentionnés ci-dessus sont compa-

Dataset	*d*	T	Task	Ratio	Sources
CMUsubject16	62	534	Action Recognition	50-50 split	Carnegie (0)
ECG	2	147	ECG Classification	50-50 split	Oleszewski (2012)
EEG	13	117	EEG Classification	50-50 split	Lichman (2013)
EEG2	64	256	EEG Classification	20-80 split	(Lichman, 2013)
KickvsPunch	62	761	Action Recognition	62-38 split	Carnegie (0)
Movement AAL	4	119	Movement Classification	50-50 split	Lichman (2013)
NetFlow	4	994	Action Recognition	60-40 split	Sübakan et al. (2014)
Occupancy	5	3758	Occupancy Classification	35-65 split	Lichman (2013)
Ozone	72	291	Weather Classification	50-50 split	Lichman (2013)
Wafer	6	198	Manufacturing Classification	25-75 split	Oleszewski (2012)
WalkVsRun	62	1918	Action Recognition	64-36 split	Carnegie (0)

Table 2.3 – Résumé des données. **T** est le nombre d'instants de temps observé, **d** : est la dimension et **Ratio** la proportion apprentissage et test des données. Toutes les bases de données sont disponible ici : https://github.com/titu1994/MLSTM-FCN/releases/tag/v1.0

rés par la précision, c'est-à-dire le taux de classes bien prédites obtenu sur l'ensemble de données de test.

2.4.1 Ajustement des hyperparamètres

Comme pour certains ensembles de données (CMUsubject, KickVsPunch, etc.), la taille de l'échantillon est petite (moins de 50 observations, voir la Table 2.4), le nombre de composantes dans MFPLS et MFPCA est choisi par validation croisée à 20-folds (contrairement à 10-folds dans les parties précédentes).

Pour toutes les méthodes d'analyse de données fonctionnelles, nous utilisons 30 fonctions de base splines par dimension afin d'obtenir une représentation fonctionnelle de chaque ensemble de données. Nous avons choisi un faible nombre de fonctions dans la base par rapport au nombre minimum de points temporels discrets (117) des ensembles de données brutes d'origine.

2.4.2 Résultats

Le Tableau 2.4 montre que, dans la plupart des cas, MFPLS et MFPCA-LDA sont compétitifs par rapport à Karim et al. (2019) et SOTA. Pour à peu près la moitié des cas, MFPLS atteint la précision la plus élevée ou la deuxième plus élevée. La principale différence entre MFPCA-LDA et MFPLS est que, dans le premier cas, les composantes sont recherchées sans tenir compte de la variable d'intérêt Y.

Datasets	N_{Train}	N_{Test}	MFPLS	MFPCA-LDA	Karim et al.	SOTA	Methods
CMUsubject16	29	29	86.21	89.66	**100**	**100**	[1]
ECG	100	100	85	88	86	**93**	[2]
EEG	64	64	48.44	46.88	**65.63**	62.5	[3]
EEG2	600	600	81.83	72.17	**91.33**	77.5	[3]
KickVsPunch	16	10	90	80	**100**	**100**	[2]
MovementAAL	157	157	67.52	61.78	**79.63**	65.61	[4]
NetFlow	803	534	84.64	80.90	95	**98**	[2]
Occupancy	41	76	71.05	**80.26**	76	67.11	[4]
Ozone	173	173	73.99	79.19	**81.5**	75.14	[5]
Wafer	298	896	85.04	97.32	**99**	**99**	[2]
WalkVsRun	28	16	**100**	**100**	**100**	**100**	[2]

Table 2.4 – Comparaison des méthodes par leurs précisions (en %) dans l'échantillon test. **[1]** : Tuncel and Baydogan (2018), **[2]** :Schäfer and Leser (2017), **[3]** :RF , **[4]** : SVM-Poly, **[5]** : DTW

2.5 Conclusion et perspectives

L'apprentissage statistique de données fonctionnelles multivariées évoluant dans des espaces complexes soulève des questions qui nécessitent le développement de nouvelles méthodologies. Dans ce chapitre, nous proposons une méthode PLS (MFPLS) pour la classification binaire et la régression lorsque les variables explicatives sont multivariées fonctionnelles, potentiellement avec des domaines différents. Les performances de cette méthode sur données réelles et simulées sont fournies, en plus des arguments techniques : relation entre FPLS et MFPLS, forme de la fonction coefficient, etc. Les expériences numériques nous ont permis de comparer MFPLS à certaines méthodologies compétitrices, en particulier un autre algorithme PLS (Beyaztas and Shang (2022)) et d'autres méthodes bien connues de la littérature ; la régression en composantes principales (MFPCA) et des algorithmes d'apprentissage automatique (RF, SVM, etc.).

Contrairement à la plupart de ces méthodes, notre approche permet la prise en compte de données fonctionnelles multivariées pouvant être définies sur des domaines différents. Cela permet de traiter simultanément plusieurs types de données (par exemple des images, des séries temporelles, etc.) avec un nombre potentiellement important d'applications, comme le montre l'étude de cas de classification sur des images et des séries temporelles fonctionnelles. Cette dernière montre aussi que lorsque les données sont bruitées, MFPLS surpasse la méthode de discrimination linéaire reposant sur les scores de l'analyse en composante principale (MFPCA-LDA).

En général, les expériences numériques montrent la compétitivité de la méthode proposée par rapport à certaines méthodes existantes. Les applications sur la classification de séries temporelles multivariées mettent en évidence les performances compétitives de MFPLS par rapport à des modèles black-box (LSTM, RF, etc). Ces performances peuvent être améliorées en utilisant des connaissances contextuelles sur les données (groupes de variables, prétraitement approprié, etc).

L'ajout de ces informations pourra être exploré à l'avenir, ainsi que de possibles travaux autour de MFPLS. Par exemple, dans ce chapitre, nous nous sommes concentrés sur les variables fonctionnelles continues. Une extension possible de MFPLS consisterait à inclure d'autres types de covariables, par exemple, fonctionnelles qualitatives.

De plus, dans la Section 2.4, certaines des données prises en compte (EEG, Ozone, etc.) peuvent présenter une dépendance spatiale au sein des dimensions. Notre approche ne semble pas en être affectée, mais cela mérite d'être étudié à l'avenir.

D'une manière générale, dans un certain nombre de travaux sur l'analyse de données fonctionnelles, des paramètres de réglages liés au nombre de fonctions de base utilisées pour lisser les données brutes et pour réduire la dimension de l'espace fonctionnel doivent être sélectionnés. Ici, nous avons soit fixé ces paramètres, soit utilisé une approche de validation croisée pour les déterminer. D'autres possibilités peuvent être construites à partir de méthodes bootstrap ou de critères tels que AIC, BIC.

Dans le prochain chapitre, nous nous concentrons sur deux autres limitations de l'approche MFPLS. La première est l'hypothèse de linéarité entre X et Y. En effet, lorsque les données sont hétérogènes et à haute dimension, il est possible que l'approche MFPLS soit inefficace. La deuxième limitation est que l'approche présentée ne peut être utilisée que pour la classification binaire. Dans le chapitre suivant, nous présentons un arbre de décision qui permet la classification à plus de deux classes. Il est d'une certaine manière une extension plus flexible de l'approche MFPLS.

Annexes

.1 Preuves

Proposition 1. Ici C-S désigne l'inégalité de Cauchy-Schwartz pour les intégrales (1) et sur les sommes (2).

$$\begin{aligned}
&\mathrm{Cov}^2(\langle X, w\rangle_{\mathcal{H}}, Y) = \mathbb{E}^2\left(\langle X, w\rangle_{\mathcal{H}} Y\right) \\
&\qquad\qquad = \left[\sum_{j=1}^{p}\left[\int_{\mathcal{T}_j}\mathbb{E}\left(X^{(j)}(t)Y\right)w^{(j)}(t)dt\right]\right]^2 \\
\text{C-S(1)} \implies \mathrm{Cov}^2(\langle X, w\rangle_{\mathcal{H}}, Y) &\leq \left[\sum_{j=1}^{p}\left(\int_{\mathcal{T}_j}\mathbb{E}^2(X^{(j)}(t)Y)dt\right)^{1/2}\right. \\
&\qquad\qquad \left.\left(\int_{\mathcal{T}_j}[w^{(j)}(t)]^2dt\right)^{1/2}\right]^2 \\
\text{C-S (2)} \implies \mathrm{Cov}^2(\langle X, w\rangle_{\mathcal{H}}, Y) &\leq \left[\sum_{j=1}^{p}\int_{\mathcal{T}_j}\mathbb{E}^2(X^{(j)}(t)Y)dt\right]\underbrace{\left[\sum_{j=1}^{p}\int_{\mathcal{T}_j}[w^{(j)}(t)]^2dt\right]}_{||w||^2_{\mathcal{H}}=1} \\
&\mathrm{Cov}^2(\langle X, w\rangle_{\mathcal{H}}, Y) \leq \sum_{j=1}^{p}\int_{\mathcal{T}_j}\mathbb{E}^2(X^{(j)}(t)Y)dt
\end{aligned}$$

Les inégalités C-S deviennent des égalités, c'est-à-dire que les maximums sont atteints, si pour $j = 1, \ldots, p$ il existe des scalaires non-nuls a et a' tels que :

- $w^{(j)}(t) = a\mathbb{E}(X^{(j)}(t)Y), t \in \mathcal{T}_j$
- $\left(\int_{\mathcal{T}_j}[w^{(j)}(t)]^2dt\right)^{1/2} = a'\left(\int_{\mathcal{T}_j}\mathbb{E}^2(X^{(j)}(t)Y)dt\right)^{1/2}$.

La première condition implique la seconde.

En effet, si $w^{(j)}(t) = a\mathbb{E}(X^{(j)}(t)Y)$ alors $\left(\int_{\mathcal{T}_j}[w^{(j)}(t)]^2dt\right)^{1/2} = |a|\left(\int_{\mathcal{T}_j}\mathbb{E}^2(X^{(j)}(t)Y)dt\right)^{1/2}$.

Autrement dit, cela revient à $a' = |a|$.

Afin d'avoir $||w||_{\mathcal{H}} = 1$, on prend $a = \left(\sum_{j=1}^{p}\int_{\mathcal{T}_j}\mathbb{E}^2(X^{(j)}(t)Y)dt\right)^{-1/2}$.

Finalement, la solution de (2.6) est

$$w^{(j)}(t) = \frac{\mathbb{E}(X^{(i)}(t)Y)}{\left(\sum_{j=1}^{p}\int_{\mathcal{T}_j}\mathbb{E}^2(X^{(j)}(t)Y)dt\right)^{1/2}},\ t \in \mathcal{T}_j. \tag{16}$$

□

Proposition 2. Le résidu d'ordre 1 (X_1) de X est donnée par $X = \xi_1\rho_1 + X_1$, où X_1 vérifie

$$\mathbb{E}(\xi_1 X_1) = 0_{\mathbb{R}^d} \iff \mathbb{E}(\xi_1 X_1^{(j)}(t)) = 0\ t \in \mathcal{T}_j,\ 1 \leq j \leq d. \tag{17}$$

De la même manière, les résidus d'ordre supérieurs vérifient aussi

$$\mathbb{E}(\xi_h X_h) = 0_{\mathbb{R}^d} \; \forall h \in \mathbb{N}^*. \tag{18}$$

Pour montrer que $\{\xi_k\}_{k=1}^h$ forme un système orthogonal, nous utilisons une démonstration par récurrence, comme dans Tenenhaus et al. (1995).

Pour $h = 0$, l'hypothèse est vérifiée. En effet, (17) implique que

$$\mathbb{E}(\xi_1 \xi_2) = \sum_{j=1}^{p} \int_{\mathcal{T}_j} \mathbb{E}\left(\xi_1 X_1^{(j)}(t)\right) w_2^{(j)}(t) dt = 0.$$

Supposons que l'hypothèse d'induction : $\mathcal{H}_0$, $\mathcal{H}_0$: $\{\xi_k\}_{k=1}^h$ *forme un système orthogonal* $h \geq 1$ est vraie

$$\begin{aligned}
&\mathbb{E}(\xi_h \xi_{h+1}) = \sum_{j=1}^{p} \int_{\mathcal{T}_j} \mathbb{E}\left(\xi_h X_h^{(j)}(t)\right) w_{h+1}^{(j)}(t) dt \\
(18) \implies &\mathbb{E}(\xi_h \xi_{h+1}) = 0 \\
&\mathbb{E}(\xi_{h-1} \xi_{h+1}) = \sum_{j=1}^{p} \int_{\mathcal{T}_j} \mathbb{E}\left(\xi_{h-1} X_h^{(j)}(t)\right) w_{h+1}^{(j)}(t) dt \\
\text{Since } X_{h-1} = \rho_h \xi_h + X_h& \\
\implies &\mathbb{E}(\xi_{h-1} \xi_{h+1}) = \sum_{j=1}^{p} \int_{\mathcal{T}_j} \underbrace{\mathbb{E}\left(\xi_{h-1} X_{h-1}^{(j)}(t)\right)}_{=0 \text{ by } (18)} dt \\
&\qquad - \rho_h^{(j)}(t) \int_{\mathcal{T}_j} \underbrace{\mathbb{E}\left(\xi_{h-1} \xi_h\right)}_{=0 \text{ by } \mathcal{H}_0} \sum_{j=1}^{p} w_{h+1}^{(j)}(t) dt, \\
\text{then } &\mathbb{E}(\xi_{h-1} \xi_{h+1}) = 0
\end{aligned}$$

La même procédure peut être utilisée pour montrer que $\mathbb{E}(\xi_j \xi_{h+1}) = 0 \; \forall j \leq h-2$. Ainsi, $\{\xi_k\}_{k=1}^h$ forme un système orthogonal $\forall h \geq 1$.
Les formules d'expansion sont des implications de ce point. □

Lemme 1 . Pour $h = 1$, $v_1 = w_1$, comme $\xi_1 = \langle X, w_1 \rangle_{\mathcal{H}}$, l'étape d'initiation se vérifie.
Supposons que $\langle X, v_j \rangle_{\mathcal{H}} = \xi_j$ est vrai jusqu'à l'ordre h ($\forall j \leq h$).
Rappelons que,

$$\xi_{h+1} = \langle X_h, w_{h+1} \rangle_{\mathcal{H}}. \tag{19}$$

La seconde équation de la Proposition 2 donne :

$$X_h = X - \sum_{i=1}^{h} \rho_i \langle v_i, X \rangle_{\mathcal{H}}$$

.
Ainsi

$$\xi_{h+1} = \langle X, w_{h+1}\rangle_{\mathcal{H}} - \sum_{i=1}^{h}\langle v_i, X\rangle_{\mathcal{H}}\langle \rho_i, w_{h+1}\rangle_{\mathcal{H}} = \langle X, \underbrace{w_{h+1} - \sum_{i=1}^{h}\langle \rho_i, w_{h+1}\rangle_{\mathcal{H}} v_i}_{v_{h+1}}\rangle_{\mathcal{H}}$$

Ce qui conclut la démonstration.

□

Chapitre 3

Arbre de décision PLS pour la classification de données fonctionnelles multivariées

Ce chapitre traite de la classification (supervisée) de données fonctionnelles multivariées. Il présente un arbre de décision (TMFPLS) permettant la classification en classes multiples. TMFPLS utilise, pour scinder les nœuds, l'analyse discriminante obtenue par l'approche MFPLS. Cela permet à cet arbre de décision de prendre pleinement en compte de l'aspect fonctionnel des données, en fournissant des fonctions de coupures à chaque scission. De plus, le chapitre aborde le problème de l'ajustement des hyperparamètres nécessaires à une estimation optimale de TMFPLS. Plusieurs stratégies sont proposées. Certaines d'entre elles, implémentées durant les expériences numériques sur données simulées et données réelles, montrent que, TMFPLS s'avère être une méthode flexible qui fournit des résultats compétitifs par rapport à certaines méthodologies bien connues de la littérature.

3.1 Introduction

De plus en plus de données massives, telles que des séries temporelles et des images, proviennent de domaines sensibles (médecine, biologie, etc). D'un côté, les méthodes de *deep learning* ont montré leur efficacité pour l'apprentissage supervisé avec ce type de données, comme le montre, par exemple, l'article de synthèse Ismail Fawaz et al. (2019). De l'autre, l'utilisation de ces méthodes "black box" (qualifié ainsi en raison de leur opacité) est évité dans les domaines dans lesquels les décisions peuvent avoir des conséquences vitales, comme l'illustre Petch et al. (2022).

Ainsi, cela suscite un intérêt dans le développement de procédures flexibles et transparentes, en particulier dans le cas de problématiques supervisées.

Les séries temporelles et les images, en supposant qu'elles soient les observations discrètes de variables fonctionnelles continues, peuvent être considérées comme des données fonctionnelles. Dans ce contexte, les méthodes supervisées développées en analyse de données fonctionnelles se présentent comme une alternative aux modèles "black box" et sont au cœur des préoccupations de ce chapitre.

Plus précisément, ici, nous nous intéressons à la prédiction d'une variable catégorielle Y à partir d'une variable fonctionnelle p-multivariée $X = (X^{(1)}, \ldots, X^{(p)})^\top$. La variable Y peut prendre $K \geq 2$ modalités, notées $\{0, \ldots, K-1\}$ sans perte de généralités.

Les réalisations de la variable explicative X sont par définition dans un espace de fonction $\mathcal{H}$, que nous supposons hilbertien comme dans le précédent chapitre. Il peut être défini de manière flexible, autorisant l'intégration de données de types différents (séries temporelles et images par exemple) dans l'analyse. Bien que le cadre de domaines différents soit favorable à une variété d'applications, peu de travaux s'y sont intéressés. La classification à partir d'une variable fonctionnelle X a été principalement traitée suivant deux configurations : (1) X univariée (c.-à-d. $p = 1$), ou (2) X multivarié, les dimensions de X sont définies sur le même domaine.

Dans le cadre univarié, plusieurs modèles non paramétriques (voir par exemple Ferraty and Vieu (2006), Biau et al. (2005), Kara et al. (2017)) ont été proposés. L'interprétation des résultats obtenus par ces derniers peut se révéler plus difficile que pour les modèles paramétriques. En particulier, les méthodes linéaires sont reconnues pour être favorable à

une meilleure transparence des résultats. Elles ont été le cœur de plusieurs investigations et se sont révélées être des méthodes performantes pour la classification dans un certain nombre de cas (Delaigle and Hall, 2012a). Parmi elles, on peut citer l'analyse discriminante linéaire fonctionnelle (voir par ex. Preda et al. (2007), Hall et al. (2001), James and Hastie (2001)) et la régression logistique (voir par ex. Escabias et al. (2007)).

La seconde configuration (2) considère X comme une variable multivariée fonctionnelle. Toutefois, ses dimensions ($X^{(j)}$, $j = 1, \ldots, p$) sont définies sur le même espace de fonction. Dans ce cadre, les auteurs dans Górecki et al. (2015) proposent l'utilisation de modèles de régression pour la classification. Ils utilisent des bases de fonctions orthogonales pour la reconstruction fonctionnelle des données. Des analyses discriminantes ont aussi été proposées dans Gardner-Lubbe (2021). Ces analyses reposent sur une discrétisation du rapport des variances inter-classe et extra-classe, qui est une fonction (continue) dans ce cas.

Pour des variables explicatives fonctionnelles multivariées (potentiellement de domaines différents), l'algorithme des moindres carrées partielles (PLS) proposé dans le chapitre précédent peut être utilisé pour la classification binaire ($K = 2$).

Cependant, l'interprétabilité des modèles linéaires est mise à mal face à la grande dimension de certaines données fonctionnelles. En effet, même dans le cas de la régression linéaire, plus le nombre de dimensions augmente, plus l'interprétation de la fonction coefficient devient difficile. Dans Godwin (2013), la méthode proposée est une extension du modèle logistique au cas multivarié fonctionnel intégrant une pénalité lasso. Elle a pour but de découvrir un nombre réduit de dimensions contribuant au modèle.

Lorsque l'hypothèse linéaire n'est pas garantie, les arbres de décision permettent de fournir des modèles flexibles et interprétables. Dans Belli and Vantini (2022), un arbre de décision fonctionnel est proposé. Le partitionnement des nœuds est fait à l'aide d'un programme d'optimisation flexible (voir Section 1.3.2). D'autres arbres de décision basés sur les distances (Möller and Gertheiss (2018)), ou les réductions de dimensions (El Haouij et al. (2019), Febrero-Bande et al. (2017)) ont aussi été proposés dans la littérature. Néanmoins, ces méthodes ont été proposées soit pour le cas univarié, soit le cas multivarié avec un seul domaine, ce qui peut être restrictif dans les applications.

Dans ce chapitre, nous introduisons un nouvel arbre de décision (TMFPLS) capable de prendre en compte des variables explicatives fonctionnelles multivariées possiblement définies sur des domaines différents. Il s'inspire du modèle TPLDA présenté dans Poterie et al. (2019) pour le cas classique de données multivariées. Comme ce dernier, il peut intégrer des groupes de dimensions. À chaque nœud, la partition en sous-noeuds s'obtient à l'aide d'une procédure d'analyse discriminante linéaire. Outre la différence du type de données en entrée, les méthodes TPLDA et TMFPLS diffèrent par la procédure utilisée pour scinder les nœuds et par le nombre toléré de modalités de Y. Le modèle TPLDA emploi l'analyse discriminante pénalisée pour le partitionnement et TMFPLS l'analyse discriminante basée sur le PLS (ou PLS-DA). De plus, le modèle que nous proposons permet la classification à K classes (avec $K \geq 2$), là où TPLDA a été présenté que pour le cas binaire ($K = 2$).

L'arbre TMFPLS prend pleinement en compte l'aspect fonctionnel des données explicatives. La possibilité d'intégrer des groupes de dimensions permet de faciliter l'interprétation

des modèles. Ils ne sont pas forcément disjoints, contrairement au groupe lasso (Godwin, 2013) et, d'une certaine manière, généralisent la notion de dimension.

Le chapitre est composé de quatre parties. La première présente la méthode TMFPLS et soulève le problème de l'ajustement des hyperparamètres. Quelques stratégies proposées pour y remédier sont également discutées. La deuxième section propose des applications numériques sur données simulées. L'arbre TMFPLS est comparé à des méthodes concurrentes dans le cadre de la classification binaire. Ensuite, TMFPLS est appliqué à un cas de classification (à quatre classes) des données *Epilepsy*. Le chapitre se conclut par une discussion sur des axes de recherches futures en Section 3.5.

3.2 Méthode

Soit le couple (X, Y), où Y une variable catégorielle à valeur dans $\{0, \dots, K-1\}$ et $X = (X^{(1)}, \dots, X^{(p)})^\top$ une variable p-multivariée fonctionnelle prenant ses valeurs dans l'espace fonctionnel de Hilbert $\mathcal{H}$. Sans perte de généralités, dans la suite du chapitre, on supposera que X est de moyenne nulle.
Le produit scalaire qui équipe $\mathcal{H}$ est noté $\langle , \rangle_{\mathcal{H}}$ et est donné par :

$$\langle f, g \rangle_{\mathcal{H}} = \sum_{j=1}^{p} \int_{\mathcal{T}_j} f^{(j)}(t) g^{(j)}(t) dt$$

où $f = \left(f^{(1)}, \dots, f^{(p)}\right)^\top$ et $g = \left(g^{(1)}, \dots, g^{(p)}\right)^\top$ sont des fonctions dans $\mathcal{H}$. Pour toute fonction $f \in \mathcal{H}$, on admet que

$$||f^{(j)}||_{L_2(\mathcal{T}_j)} = \left(\int_{\mathcal{T}_j} \left(f^{(j)}(t)\right)^2 dt \right)^{1/2} < \infty$$

pour $j = 1, \dots, p$.

On suppose que la variable X est composée de $g \geq 1$ groupes (pas forcément disjoints) : $\mathcal{G}_1, \dots, \mathcal{G}_g$, où

$$\mathcal{G}_j \subset \{1, \dots, p\} \text{ , avec } j = 1, \dots, g.$$

Chaque groupe de dimensions de X, noté $X^{\mathcal{G}_j}$ où $j = 1, \dots, g$, prend valeurs dans l'espace $\mathcal{H}_{\mathcal{G}_j}$. Le produit scalaire qui équipe ce dernier est noté $\langle , \rangle_{\mathcal{G}_j}$.

En pratique, si X ne contient pas de structure de groupe, les dimensions de X peuvent être utilisées, c'est-à-dire $X^{\mathcal{G}_k} = X^{(k)}$, avec $k = 1, \dots, p$.

Cette structure de groupe est pratique, car elle facilite l'intégration des principales composantes de la variable X et, lorsque $g < p$, elle aide à l'estimation de la méthode proposée – comme nous le verrons dans la section suivante.

Dans la suite du chapitre, nous considérons l'ensemble d'apprentissage suivant : $(x_i, y_i)_{i=1,\dots,n}$. Il représente n réalisations indépendantes du couple (X, Y).

3.2.1 L'étape de segmentation

À la Section 1.3.2, nous avons évoqué la difficulté liée à la construction d'un arbre de décision pour des données fonctionnelles. En effet, pour chaque dimension, la continuité de la variable fonctionnelle univariée rend infaisable la recherche (exhaustive) d'un seuil de rupture optimal. L'arbre de décision proposé dans cette section utilise la régression des moindres carrées partielles (PLS), présentée dans le chapitre précédent, comme un moyen de contourner cette difficulté.

La règle utilisée pour scinder un nœud repose sur l'analyse discriminante. Cette stratégie a été utilisée dans le cadre multivarié classique par le modèle TPLDA (Poterie et al., 2019), où l'analyse discriminante pénalisée linéaire sert aux partitionnements successifs des nœuds dans l'arbre. Dans le cadre fonctionnel, l'analyse discriminante linéaire est un problème inverse mal posé (Preda et al., 2007). Lorsque $K = 2$ (classification binaire), elle peut être adressée avec la régression PLS, comme déjà mentionnée dans le chapitre précédent. L'utilisation de la régression dans ce cadre repose sur un codage pratique de la variable d'intérêt Y. En effet, une équivalence existe entre la régression linéaire et l'analyse discriminante linéaire lorsque $K = 2$; pour plus de détails, voir la Section 2.2.3.

L'arbre de décision TMFPLS repose sur cette technique. Bien que Y puisse prendre plus de deux modalités ($K \geq 2$), l'analyse discriminante linéaire pour la classification binaire sert à scinder les nœuds. Comme TPLDA, TMFPLS est binaire. L'objectif des arbres binaires est de diviser un nœud en deux sous-nœuds (*child nodes*) plus purs, c'est-à-dire avec des répartitions plus homogènes des classes. Dans notre cas, cette étape est réalisée à l'aide, en plus de l'analyse discriminante linéaire, des variables indicatrices associées aux classes de Y $(0, \ldots, K-1)$.

Plus précisément, considérons un nœud $\mathcal{C}$ contenant les observations $(x_i, y_i)_{i \in \mathcal{C}}$. L'algorithme TMFPLS se compose de deux parties.

Étape 1 : Les partitionnements candidats

Pour $j = 1, \ldots, g$ et $k = 0, \ldots, K-1$:

— Faire l'analyse discriminante (MFPLS) à l'aide des données $(x_i^{\mathcal{G}_j}, z_i^k)_{i \in \mathcal{C}}$,

où

$$z_i^k = \begin{cases} 1 & \text{si } y_i = k \\ 0 & \text{sinon.} \end{cases}$$

— Le modèle estimé, noté $\mathcal{M}_{j,k} : \mathcal{H}_{\mathcal{G}_j} \to \mathbb{R}$, scinde $\mathcal{C}$ en deux nœuds $\mathcal{C}_{j,k}^0, \mathcal{C}_{j,k}^1$,

$$\mathcal{C}_{j,k}^0 = \{i \in \mathcal{C}, \mathcal{M}_{j,k}(x_i^{\mathcal{G}_j}) \leq 0\}$$
$$\mathcal{C}_{j,k}^0 = \{i \in \mathcal{C}, \mathcal{M}_{j,k}(x_i^{\mathcal{G}_j}) > 0\}$$

où $\mathcal{M}_{j,k}(x^{\mathcal{G}_j}) = \langle x^{\mathcal{G}_j}, \beta_k^{\mathcal{G}_j} \rangle_{\mathcal{H}_{\mathcal{G}_j}}$, et $x^{\mathcal{G}_j} \in \mathcal{H}_{\mathcal{G}_j}$

Étape 2 : Le partitionnement optimal

— Trouver le couple $(\mathcal{J}, \mathcal{K})$ qui maximise la décroissance de la fonction *impureté* $\Delta\mathcal{Q}$ (Poterie et al. (2019)),

$$(\mathcal{J}, \mathcal{K}) = \underset{(j,k)\in\{1,\dots,p\}\times\{0,\dots,K-1\}}{\arg\max} \Delta\mathcal{Q}(j,k).$$

La fonction d'impureté $\Delta\mathcal{Q}$ est, dans notre cas, donnée par

$$\Delta\mathcal{Q}(j,k) = |\mathcal{C}|\mathcal{Q}(\mathcal{C}) - |\mathcal{C}^0_{j,k}|\mathcal{Q}(\mathcal{C}^0_{j,k}) - |\mathcal{C}^1_{j,k}|\mathcal{Q}(\mathcal{C}^1_{j,k}),$$

où $|\mathcal{C}|$ est le cardinal de $\mathcal{C}$ et $\mathcal{Q}(\mathcal{C})$ est l'indice de Gini au nœud $\mathcal{C}$.

— Finalement, scinder $\mathcal{C}$ en utilisant $\mathcal{M}_{\mathcal{J},\mathcal{K}}$. Autrement dit, la partition retenue est : $\mathcal{C}^0_{\mathcal{J},\mathcal{K}}$ et $\mathcal{C}^1_{\mathcal{J},\mathcal{K}}$.

Ces étapes sont effectuées sur les sous-noeuds, les sous-noeuds résultants et ainsi de suite. Des hyperparamètres tels que la profondeur maximale (m) et la pureté minimale dans les nœuds (η) servent comme critères d'arrêts à l'algorithme.

Par exemple, lorsqu'un nœud a un indice d'impureté inférieur à η, les deux étapes n'y sont pas appliquées. Ces nœuds sont jugés suffisamment "purs". Les nœuds terminaux désignent les nœuds purs et les nœuds dans lesquels aucune partition ne conduit à une diminution de l'impureté. Ces nœuds servent à la prédiction. En effet, considérons une fonction $x_0 \in \mathcal{H}$ et y_0 sa classe. Les modèles MFPLS estimés à chaque étape servent à déterminer le nœud terminal $\tilde{\mathcal{C}}$ auquel x_0 appartient. La classe la plus représentée dans $\tilde{\mathcal{C}}$ fournit une prédiction de y_0 notée $\hat{y}_0$.

Remarque 4.

— *Plus le nombre de groupes g et le nombre de classes K sont grands et plus l'apprentissage de l'arbre TMFPLS est coûteux en temps de calculs. Plus précisément, $K \times g$ modèles MFPLS sont estimés pour la partition d'un nœud. Lorsque le nombre de dimensions de la variable explicatives X est élevé, utiliser des groupes plutôt que les dimensions de X permet d'accélérer l'étape d'apprentissage.*

— *Lorsque de fortes différences sont présentes dans les effectifs des groupes, un biais de sélection en faveur des groupes avec le plus d'effectif est possible (Poterie et al., 2019). Des critères pénalisés $\Delta\mathcal{Q}$ sont alors conseillés.*

L'arbre estimé à l'aide de (x_i, y_i) pour $i = 1, \dots, n$ sans trop de contrainte (par exemple la taille de l'arbre, etc.) est dit maximal. Il peut être sujet au sur-apprentissage sur ces données. Une telle situation le rendrait inefficace à la généralisation sur d'autres exemples. Trois stratégies sont généralement utilisées (parfois simultanément) pour éviter ce problème.

1 Le *pre-pruning* consiste à rajouter un certain nombre d'hyperparamètres pour forcer l'arrêt prématuré de l'algorithme. Ceux-ci peuvent agir sur l'effectif minimal des individus dans les nœuds, la profondeur maximale de l'arbre, etc.

2 La forêt de décision est une méthode efficace pour la prédiction. Elle évite le problème du sur-apprentissage en construisant plusieurs arbres de décisions à l'aide de partitions des données disponibles pour l'apprentissage. Ces arbres sont ensuite utilisés pour la prédiction à l'aide d'un système de vote. La forêt de décision est cependant un modèle black box, dans la mesure où il est difficile d'interpréter clairement ces résultats.

3 Enfin, le *post-pruning* préconise la construction maximale de l'arbre. À l'aide d'une base de données de validation, l'objectif est de trouver l'arbre optimal. Cela est fait en testant l'apport prédictif – avec des critères performances tels que l'AUC, la précision, etc.– de certaines ramifications de l'arbre maximal.

La partie suivante présente les méthodes de *post-pruning* proposées pour éviter le sur-apprentissage de TMFPLS.

3.2.2 Stratégies contre le sur-apprentissage

De la même manière que dans Poterie et al. (2019), une stratégie de pruning basée sur la profondeur optimale de l'arbre peut être appliqué à l'arbre TMFPLS.

Dans un premier temps, supposons qu'un jeu de données de validation composé de n_v individus soit disponible. Il est noté (x_i, y_i), $i = n+1, \ldots, n+n_v$. Soit l'arbre (maximal) $\mathcal{A}_m$, de profondeur m, déterminé par l'algorithme précédent. Le modèle $\mathcal{A}_m$ est en fait composé d'une succession d'arbres imbriqués $\mathcal{A}_0, \ldots, \mathcal{A}_{m-1}$ telle que

$$\mathcal{A}_0 \subset \mathcal{A}_1 \subset \ldots \subset \mathcal{A}_{m-1} \subset \mathcal{A}_m$$

où $\mathcal{A}_0$ est l'arbre (trivial) contenant que le nœud racine. Une illustration des arbres imbriqués $\mathcal{A}_0, \ldots, \mathcal{A}_{m-1}$ est présentée dans la Figure 3.1.

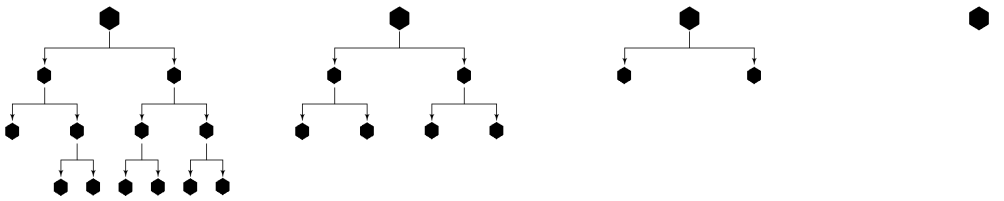

Figure 3.1 – Exemples d'arbres imbriqués

L'objectif est alors de trouver la profondeur optimale m^* telle que

$$m^* = \underset{l=0,\ldots,m}{\arg\max}\, \mathcal{R}\left[(\hat{y}^l_{n+1}, \ldots, \hat{y}^l_{n+n_v}), (y_{n+1}, \ldots, y_{n+n_v})\right], \tag{3.1}$$

où $\mathcal{R}$ est une mesure de performances (AUC, précision, etc.) et $\hat{y}^l_i$ est la prédiction de y_i fournie par l'arbre $\mathcal{A}_l$, $l = 0, \ldots, M$ et $i = n+1, \ldots, n+n_v$.

En pratique, les données de validations ne sont pas toujours disponibles et un grand nombre de composantes PLS peut aussi conduire à du sur-apprentissage. Obtenir de manière optimale ces paramètres : profondeur maximale de l'arbre et nombre de composantes

PLS, peut se révéler compliqué à réaliser par la validation croisée. Ces paramètres sont sans doute interdépendants. La taille de l'arbre pourrait être liée au nombre de composantes utilisé à chaque partitionnement. De plus, comme déjà souligné, l'estimation de l'arbre peut avoir un coût non négligeable en temps de calcul. De multiples validations croisées risquent de rendre les estimations, dans certains cas, infaisables en pratique. C'est le cas, par exemple, lorsqu'il y a un nombre élevé de classes et/ou de groupes.

Nous proposons trois stratégies pour estimer les hyperparamètres de l'arbre TFMPLS, classées de la mise en œuvre la plus coûteuse à la moins coûteuse.

Stratégie 1

Une première approche est de se concentrer uniquement sur la profondeur maximale de l'arbre. Les nombres de composantes peuvent-être fixés à priori ou choisis par des règles précises. Par exemple, ils pourraient dépendre du nombre d'individus dans les nœuds.
Cette stratégie permet de toute évidence un déploiement rapide de l'algorithme, la validation croisée n'est utilisée que pour déterminer la profondeur optimale m^* de l'arbre. Cependant, des nombres de composantes mal choisis peuvent conduire à un arbre de décision peu efficace.

Stratégie 2

Une autre méthode consiste à exiger qu'à chaque partition un même nombre fixe h de composantes soit utilisé. L'objectif reviendrait alors à trouver par la validation croisée les paramètres optimaux h^* et m^*. Estimer le nombre de composantes à l'aide des données pourrait garantir l'estimation d'un meilleur modèle. Cependant, il apparaît que dans ce cas, l'arbre perd en flexibilité. De plus, il n'existe aucune garantie permettant de justifier l'hypothèse du même nombre de composantes.

Stratégie 3

La dernière stratégie ne fait aucune hypothèse sur le nombre de composantes dans l'arbre. Elle consiste à faire des validations croisées imbriquées. La première est faite pour estimer m^*. Aucune information n'est donnée sur le nombre de composantes MFPLS. À chaque nœud, une validation croisée est effectuée pour l'estimer de manière optimale. Cette stratégie a l'avantage de donner la liberté à l'algorithme pour trouver les meilleurs hyperparamètres. Néanmoins, elle conduit rapidement–surtout lorsqu'il y a plusieurs groupes et plusieurs classes– à un coût non négligeable de temps de calculs.

3.3 Etude de simulation

Ici, nous testons le modèle TMFPLS sur des données simulées. L'objectif de cette partie est d'étudier le comportement de ce modèle pour la classification de données fonction-

nelles multivariées. Le cadre choisi des simulations fait référence au problème de reconnaissance visuelle des formes (Fukushima, 1988). Les classes sont identifiables visuellement, mais peuvent être difficiles à détecter avec des algorithmes. Ce type de problématique est rencontré, par exemple, pour la détection de pointes épileptiques dans les enregistrements d'électroencéphalogrammes (voir par ex. Abd El-Samie et al. (2018) pour plus de détails).

Dans cette étude de simulation, nous nous limitons au cas de la classification binaire. On étudie la relation entre le nombre de composantes et la profondeur de l'arbre. Les approches, introduites dans le chapitre précédent, MFPLS (pour l'analyse discriminante) et MFPCA-LDA (ici noté MFPCA pour faciliter la lecture) servent de méthodes concurrentes pour comparer les performances obtenues par TMFPLS. Comme ce dernier, ces méthodes exigent un nombre de composantes en entrée. Nous nous intéressons à l'impact du choix du nombre de composantes sur les performances de ces méthodes.

Soit une variable fonctionnelle X bivariée $X = (X^{(1)}, X^{(2)})^\top$, où la première dimension $X^{(1)}$ représente une série temporelle et la deuxième $X^{(2)}$ représente une image. La variable X prend valeur dans l'espace de fonctions $\mathcal{H} = L_2(\mathcal{T}_1) \times L_2(\mathcal{T}_2)$, où $\mathcal{T}_1 = [0, 1]$ et $\mathcal{T}_2 = \mathcal{T}_1 \times \mathcal{T}_1$. Elle est définie en utilisant les variables aléatoires discrètes $a_1, \ldots, a_4$ à valeurs dans $\{-1, 0, 1\}$ comme :

$$X^{(1)}(t) = \sum_{s=1}^{4} a_s \Delta_s^{(1)}(t) + \epsilon^{(1)}(t), \qquad t \in \mathcal{T}_1$$
$$X^{(2)}(t) = \sum_{s=1}^{4} (1 - a_s) \Delta_s^{(2)}(t) + \epsilon^{(2)}(t), \qquad t \in \mathcal{T}_2.$$

Pour $k = 1, \ldots, 4$, les fonctions $\Delta_k^{(1)}$, $\Delta_k^{(2)}$ sont des fonctions déterministes "triangles" et $\epsilon^{(1)}, \epsilon^{(2)}$ sont les fonctions résidus générées comme des bruits blancs.
Les premières sont définies comme :

$$\Delta_k^{(1)}(t) = (1 - 10|t - u_k|)_+$$
$$\Delta_k^{(2)}(s) = \left(1 - 5\left(|s^{(1)} - v_k^{(1)}| + |s^{(2)} - v_k^{(2)}|\right)\right)_+$$

où $s = (s^{(1)}, s^{(2)})$ désigne un indice dans $\mathcal{T}_2$ et $t \in \mathcal{T}_1$. La quantité $(.)_+ : \mathbb{R} \to \mathbb{R}^+$ est la fonction partie positive. Les éléments $\{u_k\}_{k=1}^4$ et $\{v_k\}_{k=1}^4$ définissent les points respectivement dans les intervalles $\mathcal{T}_1$ et $\mathcal{T}_2$ où les fonctions triangles atteignent leurs maximums. Ici, ils sont fixés et donnés par :

$$\begin{aligned} u_1 &= 0.2 & v_1 &= (0.2, 0.2) \\ u_2 &= 0.4 & v_2 &= (0.2, 0.8) \\ u_3 &= 0.6 & v_3 &= (0.8, 0.2) \\ u_4 &= 0.8 & v_4 &= (0.8, 0.8). \end{aligned}$$

Les fonctions résidus $\epsilon^{(1)}$ et $\epsilon^{(2)}$ sont générées comme des bruits blancs avec une variance égale à 0.05. Autrement dit, elles vérifient les propriétés suivantes :

$$\mathbb{E}(\epsilon^{(j)}(t)) = 0, \ \mathbb{E}(\epsilon^{(j)}(t)\epsilon^{(j)}(s)) = 0, \ \mathbb{V}(\epsilon^{(j)}(t)) = 0.05$$

pour $s \neq t$, où $s, t \in \mathcal{T}_j$ et $j = 1, 2$.

La séquence $a_1, \ldots, a_4$ peut être vue comme un vecteur aléatoire $a = (a_1, \ldots, a_4)^\top$. Ce dernier prend valeur dans $\{-1, 0, 1\}^4$. Il est facile de montrer que 81 évènements $(V_1, \ldots, V_{81})$ constituent son univers Ω_a de probabilité, $\Omega_a = \{V_1, \ldots, V_{81}\}$. Sans perte de généralités, on suppose que $V_1 = \begin{pmatrix} 1 & 1 & 0 & 0 \end{pmatrix}^\top$, $V_2 = \begin{pmatrix} 0 & 1 & 1 & 0 \end{pmatrix}^\top$, $V_3 = -V_1$ et $V_4 = -V_2$, les vecteurs $V_5, \ldots, V_{81}$ sont indifféremment les autres éléments de Ω_a.
Nous définissons la variable d'intérêt Y comme

$$Y = \begin{cases} 1 & a \in \{V_1, \ldots, V_4\} \\ 0 & \text{sinon,} \end{cases} \tag{3.2}$$

Si $a_s \neq 0$, $s = 1, \ldots, 4$, cela signifie que l'on observe un pic sur la dimension $X^{(1)}$ à la position $t = u_s$. Il peut être positif ($a_s = 1$) ou négatif ($a_s = -1$). Alors, $Y = 1$, si exactement deux pics consécutifs (négatifs ou positifs) sont présents entre 0 et 0.6 en $X^{(1)}$. Les pics d'intérêt sont observés soit à $(0.2, 0.4)$, soit à $(0.4, 0.6)$. La Figure 3.2 illustre les quatre cas $(V_1, \ldots, V_4)$ où $Y = 1$.
Deux scénarios sont étudiés suivant les probabilités d'occurrences des événements $\{V_j\}_{j=1}^{81}$.

- Scénario 1 : Les pics positifs forment la classe $Y = 1$.
  $\mathbb{P}(a = V_1) = \mathbb{P}(a = V_2) = \frac{1}{4}$
  $\mathbb{P}(a = V_3) = \mathbb{P}(a = V_4) = 0$
  $\mathbb{P}(a = V_5) = \ldots = \mathbb{P}(a = V_{81}) = \frac{1}{162}$.
- Scénario 2 : Les pics positifs et négatifs forment la classe $Y = 1$.
  $\mathbb{P}(a = V_1) = \ldots = \mathbb{P}(a = V_4) = \frac{1}{8}$
  $\mathbb{P}(a = V_5) = \ldots = \mathbb{P}(a = V_{81}) = \frac{1}{162}$.

Remarque 5.

- *Les probabilités suivant les deux scénarios de $Y = 1$ et $Y = 0$ sont les mêmes (0.5).*
- *Dans le premier scénario, les courbes X où $Y = 1$ sont composées uniquement des réalisations des événements V_1 et V_2. Elles ont en $X^{(1)}$ des pics positifs consécutifs à $(0.2, 0.4)$, ou à $(0.4, 0.6)$.*
- *Le second scénario est plus complexe. Les quatre événements $V_1, \ldots, V_4$ ont la même probabilité de se réaliser. Dans ce cas, la classe $Y = 1$ est plus hétérogène comparée au premier scénario. Les quatre cas, représentés dans la Figure 3.2, ont théoriquement la même proportion.*

Pour chaque scénario, on génère $n = 200$ réalisations indépendantes du couple (X, Y) : 80% sont utilisées pour l'apprentissage et 20% pour tester les méthodes. L'expérience est répliquée 100 fois et l'AUC sert à mesurer les performances des modèles.

La version fonctionnelle de X, pour chaque dimension, est reconstituée à l'aide d'une base spline (d'ordre 2) de 20 fonctions. Les fonctions utilisées sont univariées, pour la première dimension, et bivariées, pour la seconde. L'expansion dans les bases de fonctions est réalisée à l'aide du package Happ (2017). La Figure 3.3 représente des exemples de courbes (images et séries temporelles) avant et après le lissage.

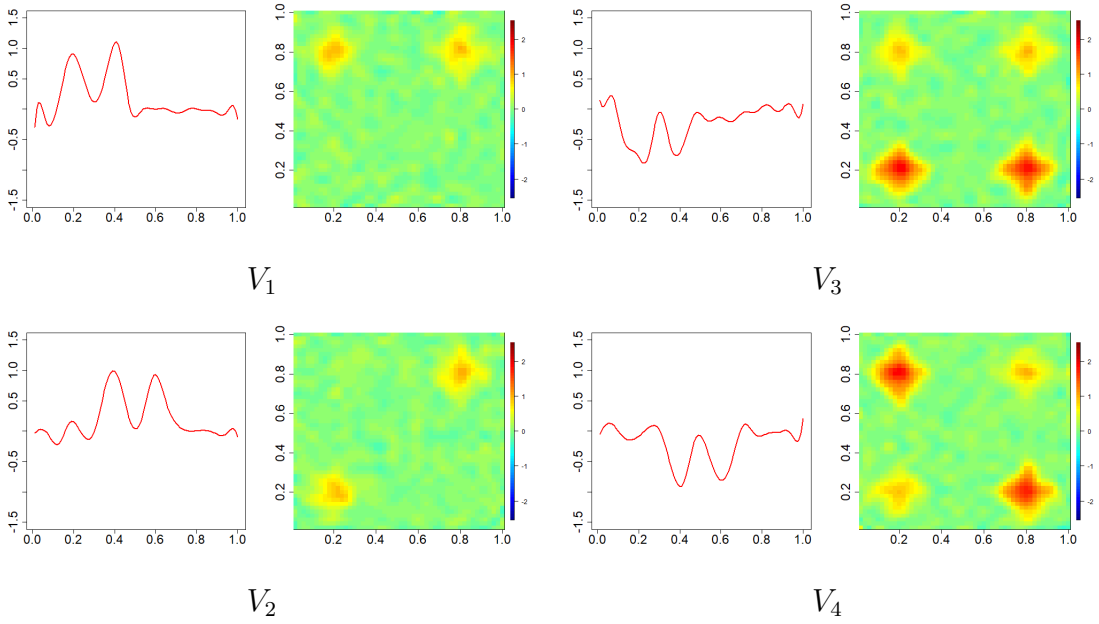

Figure 3.2 – Exemples de réalisations de X (lissées) où $Y = 1$, suivant les quatre possibilités V_1, V_2, V_3 et V_4.

Les groupes considérés dans l'arbre sont les dimensions de X, $\mathcal{G}_1 = \{1\}$ et $\mathcal{G}_2 = \{2\}$. Pour chaque méthode, les 5 premières composantes sont testées. Sachant ce nombre de composantes, à chaque estimation de TMFPLS, une validation croisée à 3-folds est utilisée pour déterminer la profondeur optimale de l'arbre TMFPLS (conformément à la stratégie 1).

3.3.1 Résultats

La Figure 3.4 présente le résumé des résultats obtenus lors des simulations. Elle donne la distribution de l'AUC (*Area under curve*) obtenue sur la base de validation et la profondeur estimée de l'arbre, suivant le nombre de composantes et le scénario.

L'intégralité des méthodes testées fournissent de bonnes performances dans le premier scénario. Les moyennes des AUC obtenus sur l'ensemble de validation sont toutes supérieures à 75%. On remarque tout de même qu'avec une seule composante, l'arbre TMFPLS et l'analyse discriminante MFPLS sont plus performants que l'analyse discriminante basée sur les scores de l'ACP (MFPCA). La légère différence de résultat entre MFPLS et MFPCA pourrait être due au fait que les composantes des moindres carrées partielles (PLS) sont cherchées de manière supervisée (c.-à-d. en utilisant Y), contrairement à ceux de l'analyse en composantes principales (ACP). L'arbre TMFPLS donne de meilleures performances que MFPLS pour une composante. La profondeur optimale, estimée en moyenne à 3, est peut-être la cause de cette différence, voir la *Profondeur estimée* du premier scénario dans la Figure 3.4. En effet, il se peut que le modèle TMFPLS utilise la profondeur de l'arbre comme un paramètre d'ajustement.

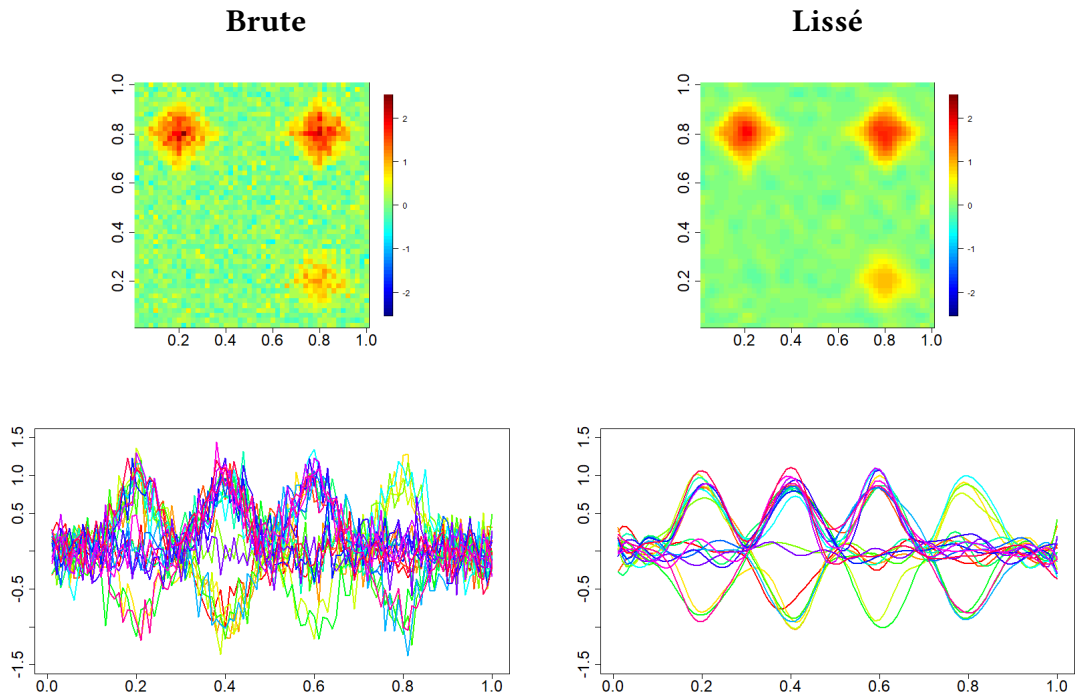

Figure 3.3 – Exemples de reconstruction fonctionnelle à l'aide de l'expansion en base de fonctions splines

Pour analyser plus en détails les performances des modèles, la Figure 3.5 représente des exemples des modèles estimés avec une seule composante. Les données qui ont servi à leurs estimations ont été choisies aléatoirement parmi les 100 expériences.

Le modèle TMFPLS présenté utilise la seconde dimension pour discriminer au mieux la classe 0. Cela a du sens, étant donné que dans le premier scénario seuls les événements V_1 et V_2 sont présents. Ainsi, lorsque des pics sont présents en $X^{(2)}$ aux points $(0.2, 0.2)$ ou $(0.2, 0.8)$, on est en présence de la classe 0. Cette seule étape permet de bien classer 80% des éléments de la classe 0. Les fonctions estimées par les modèles concurrents à l'arbre reconnaissent ces endroits de l'image comme étant significatifs pour la discrimination, en témoigne leurs fortes amplitudes en valeur absolue à ces positions. Au nœud droit à la profondeur 1, la fonction estimée n'agit que sur la première dimension de X. Par contre, à la profondeur 2 de l'arbre, les fonctions utilisées pour les ruptures se concentrent que sur la deuxième dimension de X. De cette façon, l'arbre estimé arrive à classer correctement 156 courbes sur 160, soit un pourcentage de 97.5%. Dans les faits, on voit que 4 fonctions de ruptures (à peu près 4 composantes PLS) sont utilisées pour construire l'arbre. Cependant, même avec ce nombre de composantes, la méthode MFPLS n'atteint pas ce niveau de discrimination. Cela pourrait s'expliquer par le fait que la relation entre X et Y n'est pas tout à fait linéaire.

La deuxième ligne de la Figure 3.4 montre qu'en termes d'AUC les méthodes concurrentes au TMFPLS sont inefficaces pour le second scénario. Elles donnent des performances autour 0.5, soit la répartition des classes dans les données. Il apparait aussi que plus le nombre de composantes utilisé dans l'arbre augmente et plus sa performance diminue.

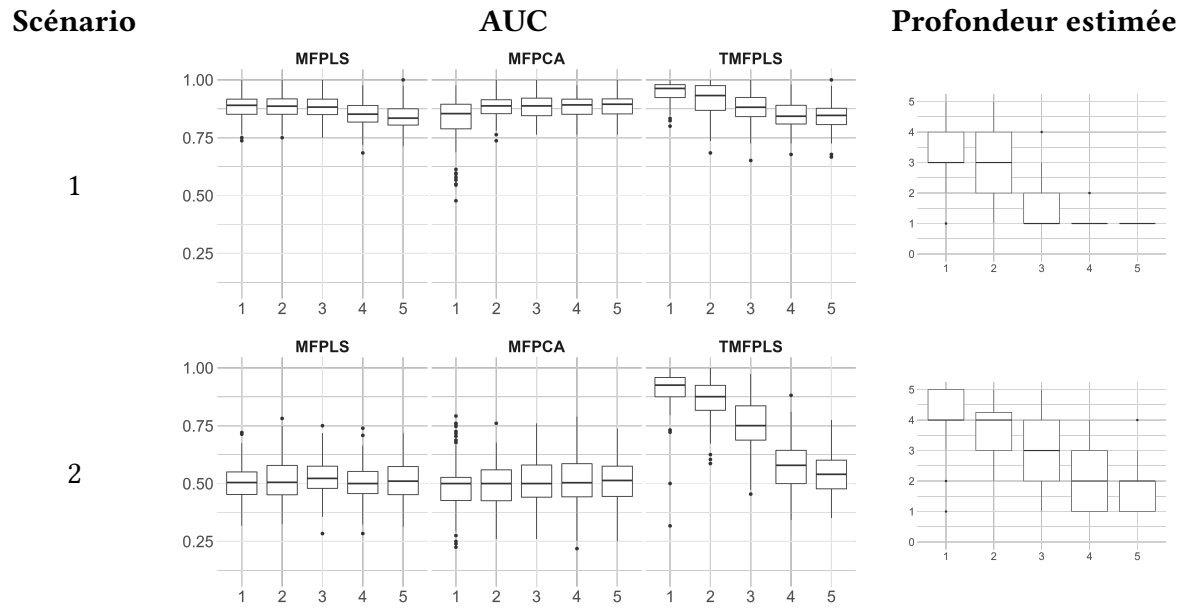

Figure 3.4 – Pour chaque scénario, les résultats (AUC et profondeur optimale de l'arbre) obtenus suivant le nombre de composantes (en abscisse), de 1 à 5

L'arbre est clairement en sur-apprentissage dans ces configurations. La dynamique de la profondeur par rapport au nombre de composantes est la même que dans le scénario précédent. Bien que la profondeur optimale (à peu près à 4) soit en moyenne plus élevée.

La Figure 3.6 donne des exemples de trois modèles estimés avec une composante. Comme précédemment, les données utilisées pour fournir ces modèles sont choisies aléatoirement parmi les 100 expériences. Le modèle TMFPLS a plus de ramification, il est plus complexe que le précédent. Pour des raisons de présentations, seules les trois premières fonctions utilisées pour la scission des nœuds sont représentées. La première, utilisée pour scinder le nœud racine, est similaire à celle estimée par le MFPLS. La pureté au sein des nœuds est recherchée en utilisant les deux dimensions. Dans cet exemple, la flexibilité de l'arbre est mise en avant. La Figure 3.4 montre quant à elle une AUC moyenne autour des 80% pour l'arbre avec une seule composante. Une modélisation de ce type peut être intéressante lorsque dans le modèle génératif des données n'est pas linéaire, mais peut être approximée par plusieurs modèles linéaires simples.

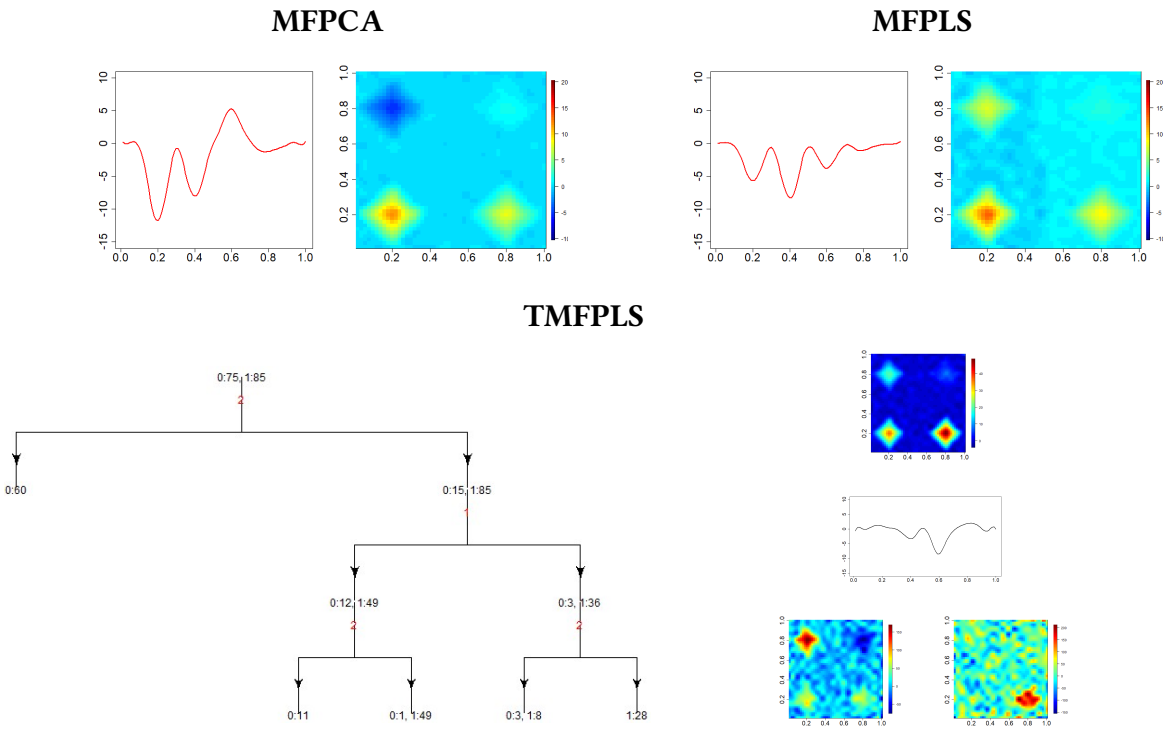

Figure 3.5 – Modèles estimés dans le scénario 1

En conclusion, cette étude de simulation a montré que l'arbre TMFPLS, bien ajusté, fournit des bons résultats sur un cas de classification visuelle. Ils sont souvent supérieurs aux modèles concurrents : MFPLS et MFPCA. Cependant, ses performances sont dépendantes des hyperparamètres suivants : la profondeur de l'arbre et les nombres de composantes PLS. En effet, cette partie a clairement démontré que le risque de sur-apprentissage vient aussi des nombres de composantes. Il semble qu'il y ait une dépendance négative entre le nombre de composantes et la profondeur de l'arbre : plus l'un augmente, plus l'autre a tendance à diminuer. Une attention particulière doit alors être portée à l'ajustement (simultané) de ces paramètres. En tenant compte de ces spécificités, la section suivante traite d'une application de la méthode proposée à des données réelles.

3.4 Application

Dans cette partie, nous testons notre méthode sur les données *Epilepsy*[1]. Elles représentent des séries temporelles multidimensionnelles et ont comme problématique la classification supervisée. Les dimensions qui composent ces données représentent les trois axes de l'espace (x, y, z). Chacune d'entre elles désigne l'enregistrement, suivant un axe spatial et en fonction du temps, de l'accélération des participants de l'expérience. Elle est enregistrée à l'aide d'un appareil placé au niveau des poignets des participants. L'objectif est de classer les quatre activités qu'ils effectuent : la marche, la course, le sciage et l'imitation

1. https://www.timeseriesclassification.com/description.php?Dataset=Epilepsy

de crise épileptiques. Les enregistrements ont une fréquence d'échantillonnage de 16 Hz et durent à peu près 13 secondes.

Les pré-traitements effectués sur ces données pour uniformiser la taille de ces enregistrements sont détaillés dans Ruiz et al. (2021). Ils conduisent à considérer 137 individus dans la base d'apprentissage (Train) et 138 dans la base de validation (Test). Les répartitions des classes sont sensiblement les mêmes, voir le Tableau 3.1.

Classes	**1**	**2**	**3**	**4**
Train	34	37	36	30
Test	34	37	37	30

Table 3.1 – Répartitions des classes dans *Epilepsy*, où **1** **2**, **3** et **4** sont respectivement les classes : imitation de crise épileptiques, marche, course et sciage.

Plusieurs méthodes, dont certaines issues du *deep-learning*, ont été testées sur ces données et ont démontré d'excellentes performances (Ruiz et al., 2021). En particulier, le modèle HIVE-COTE (Bagnall et al., 2020) prédit correctement les classes de la totalité des individus qui composent les données tests. HIVE-COTE est un méta-modèle. Il est la combinaison de plusieurs modèles de classification de séries temporelles (*Shapelet Transform Classifier*, *Time Series Forest*, etc.). Globalement, sa prédiction est issue d'une procédure de vote des modèles qui le compose.
Un autre modèle, basé sur la distance *dynamic time warping* (DTW), concurrence Hive-Cote sur ces données. Il obtient un taux de bien-classé à 96.30% sur l'ensemble test.
Bien que ces deux modèles fournissent de très bonnes performances, une interprétation claire de leurs résultats n'est pas directe, particulièrement pour le premier.
Dans la suite, leurs résultats sont comparés avec ceux obtenus par TMFPLS et l'analyse discriminante basée sur les scores de l'ACP fonctionnelle (MFPCA).

3.4.1 Classification de séries temporelles

Une première approche (peut-être la plus naturelle) consiste à considérer les données telles qu'elles, c'est-à-dire sans transformations préalables. Dans ce cas, les données explicatives sont les réalisations d'une variable fonctionnelle $X = (X^{(1)}, X^{(2)}, X^{(3)})^\top$ où $X^{(j)}$, $j = 1, 2, 3$, à valeur dans $L_2([0, 13])$, l'espace des fonctions de carrées intégrables de $[0, 13]$ à $\mathbb{R}$. Les dimensions de X sont alors définies sur le même domaine.

À chaque dimension, nous utilisons une base spline (d'ordre 3) composée de 20 fonctions pour reconstruire la version fonctionnelle des observations de X. Des exemples de lissages sont donnés dans la Figure 3.7. Ils montrent que 20 fonctions suffisent à capturer les bases fréquences des courbes (figures du haut), mais peinent à modéliser les hautes fréquences des données (figures du bas). Cependant, comme un grand nombre de fonctions peut conduire à prendre en compte le bruit des enregistrements, nous nous limitons à 20 fonctions dans cette application.

Quatre groupes de variables sont considérés dans le modèle TMFPLS : $\mathcal{G}_1 = \{1\}$, $\mathcal{G}_2 =$

$\{2\}$, $\mathcal{G}_3 = \{3\}$ et $\mathcal{G}_4 = \{1, 2, 3\}$. Cela revient à tester si les dimensions individuelles permettent une meilleure séparation que toutes les dimensions regroupées. De la même manière que les approches *sparses* (voir par exemple Godwin (2013)), l'objectif est d'avoir un minimum de dimensions contribuant aux modèles afin d'en faciliter l'interprétation.

Pour l'analyse discriminante basée sur les scores de l'ACP (MFPCA) le nombre de composantes est choisi par une validation croisée à 10-folds. Comme déjà mentionné, les performances de TMFPLS dépendent fortement des paramètres suivants : profondeur de l'arbre et le(s) nombre(s) de composantes utilisé(s) pour scinder les nœuds. Ces deux paramètres sont interdépendants. Afin de les estimer au mieux (avec un coût raisonnable de calculs), nous utilisons la stratégie 2. Autrement dit, nous exigeons que le nombre de composantes soit le même pour toutes les fonctions de coupures, et à l'aide de la validation croisée, nous déterminons simultanément ce paramètre ainsi que la profondeur optimale de l'arbre. La stratégie est mise en œuvre également à l'aide d'une validation croisée à 10-folds.

Les taux de bien classés obtenus dans l'ensemble test sont reportés dans la Table 3.2.

MFPCA	55.07%
TMFPLS	77.54%
DTW	96.30%
HIVE-COTE	100.00%

Table 3.2 – Précisions obtenues par modèles sur l'ensemble test

Les deux méthodes d'analyse fonctionnelle (MFPCA et TMFPLS) donnent des performances moindres par rapport à HIVE-COTE et DTW. On observe tout de même une différence marquée entre MFPCA et TMFPLS. L'arbre TMFPLS obtient un meilleur score. Cela pourrait être expliqué par la grande flexibilité de l'arbre estimé, représenté par la Figure 3.8. Il n'est pas aisé d'interpréter les fonctions de coupures dans une structure aussi complexe. Toutefois, on peut visuellement déterminer l'importance des groupes dans la discrimination.

3.4.2 Classification d'images et séries temporelles : temps-fréquence

Certains modèles de *deep-learning* ont la capacité de sélectionner des bandes de fréquences d'intérêt dans le signal grâce aux couches de convolution. Cette habilité leur permet une discrimination double : à la fois temporelle et fréquentielle. Les modèles d'analyse fonctionnelle, tels que présentés dans la partie précédente, sont clairement désavantagés. Ils ne peuvent influer que sur l'aspect temporel. En effet, les modèles TMFPLS et MFPCA recherchent des fonctions qui ne dépendent que du temps pour discriminer les classes.

Toutefois, il est possible d'intégrer le plan fréquentiel dans ces méthodes. Ici, nous proposons l'utilisation du spectrogramme (ou sonogramme). Une fois calculé, celui-ci se présente comme une image avec en abscisse le temps et en ordonnée les fréquences. Un pixel à la position (t, w) désigne alors l'amplitude (ou puissance) de la fréquence w à l'instant t.

De cette façon, on peut rajouter trois nouvelles dimensions $X^{(4)}$, $X^{(5)}$ et $X^{(6)}$ qui sont respectivement les spectrogrammes de $X^{(1)}$, $X^{(2)}$ et $X^{(3)}$. Leur ensemble de définition est $[0, 13] \times [0, 6.5]$. Ils sont calculés à l'aide du package *signal*[2] en utilisant une fenêtre de *hamming* de $0.62s$ et une proportion de chevauchement de 50%. La version fonctionnelle de ces nouvelles dimensions est reconstituée à l'aide d'une base composée de 5 fonctions splines bivariées, voir la Figure 3.9.

Pour l'arbre, les groupes de variables suivants sont considérés : $\mathcal{G}_1 = \{1, 4\}, \mathcal{G}_2 = \{2, 5\}$, $\mathcal{G}_3 = \{3, 6\}$, $\mathcal{G}_4 = \{1, 2, 3\}$ et $\mathcal{G}_5 = \{4, 5, 6\}$. Les premiers, $\mathcal{G}_1, \mathcal{G}_2$ et $\mathcal{G}_3$ représentent les informations temporelles et fréquentielles apportées par les axes spatiaux. Les groupes $\mathcal{G}_4$ et $\mathcal{G}_5$ sont les ensembles respectivement des composantes séries temporelles et des spectrogrammes. Une telle définition des groupes a pour but de faciliter l'interprétation du modèle.

La validation croisée à 10-folds sert à déterminer les hyperparamètres des modèles, comme dans la partie précédente.

Les taux de bien classés dans l'ensemble test obtenus par les modèles sont reportés dans la Table 3.3.

MFPCA	94.20%
TMFPLS	94.92%
DTW	96.30%
HIVE-COTE	100.00%

Table 3.3 – Précisions obtenues par modèles sur l'ensemble test

Les nouveaux résultats montrent que l'intégration des spectrogrammes permet d'améliorer significativement les résultats des modèles fonctionnels. C'est le cas surtout du modèle MFPCA où l'on observe une augmentation de 40% par rapport aux performances précédentes. Dans ces circonstances, les modèles sont certes en deçà de leurs concurrents (HIVE-COTE et DTW) mais sont compétitifs. La fonction discriminante estimée par MFPCA, représentée dans la Figure 3.10, est relativement facile à interpréter. Elle donne simultanément l'importance des composantes temporelles et fréquentielles. L'arbre TMFPLS estimé est plus simple à interpréter que le précédent, voir la Figure 3.11. Il donne un résultat légèrement supérieur à celui du MFPCA. La scission du nœud racine est faite uniquement grâce aux informations de la seconde dimension. Comme précédemment, l'utilisation des groupes permet de se faire une idée sur l'importance des groupes de variables utiles à la discrimination. On voit par exemple que le groupe le plus utilisé pour scinder les nœuds est celui composé des spectrogrammes des trois dimensions $\{4, 5, 6\}$, d'où l'importance d'inclure les fréquences dans l'analyse.

2. https://cran.r-project.org/web/packages/signal/signal.pdf

3.5 Conclusion et perspectives

Dans ce chapitre, nous avons présenté l'arbre TMFPLS. Les applications numériques montrent qu'il peut surpasser les modèles d'analyse discriminante linéaire basée sur l'analyse en composante principale (ACP) et sur la régression des moindres carrées partielles (MFPLS). La différence entre ces approches est d'autant plus marquée lorsque la relation entre la variable explicative X et la variable d'intérêt Y n'est pas tout à fait linéaire. Tout comme ses modèles concurrents, il a l'habilité d'intégrer dans la classification des séries temporelles et des images simultanément. La Section 3.4 montre que ce cadre permet la prise en compte explicite des fréquences dans la classification, favorisant la compétitivité face aux modèles black box.

L'arbre TMFPLS dépend de deux types d'hyperparamètres : les nombres de composantes PLS des fonctions de coupures et la profondeur optimale de l'arbre. La Section 3.3 a montré que mal spécifié, ils influent négativement sur la performance de l'arbre. Nous avons trouvé la stratégie 2, consistant à fixer le même nombre de composantes dans les nœuds et optimiser la profondeur optimale, comme étant une technique facile à mettre en œuvre et fournissant de bons résultats. Cependant, pour la totalité des cas, le nombre de composantes choisi pour les nœuds a été 1. Ainsi, cela revient à n'utiliser que la profondeur optimale de l'arbre comme paramètre d'ajustement dans l'apprentissage, voir la Figure 3.4. Il se pourrait que la stratégie 3 fournisse des arbres avec moins de ramifications et tout aussi performant. Cependant, son coût de temps de calculs complique son implémentation. Des travaux futurs auront pour but d'étudier sa mise en œuvre optimale.

Ici, nous nous sommes intéressés à la classification. Une extension de l'arbre TMFPLS à des fins de régression peut être explorée dans des recherches futures. Une telle entreprise comprend comme principale difficulté la règle de partitionnement des nœuds. L'analyse discriminante MFPLS est un outil désigné pour la classification, mais inefficace pour la régression. L'objectif d'un arbre (binaire) pour la régression serait de diviser un nœud en deux sous-nœuds les plus homogènes possibles dans le sens de leurs fonctions coefficients, c'est-à-dire ayant la même relation entre X et Y. Cela revient à rechercher dans le nœud deux classes latentes correspondantes chacune à un modèle différent. Une approche basée sur la régression par *clusters*, telle que proposée dans Preda and Saporta (2005), semble adéquate à ce but. Cependant, une attention particulière doit être portée sur la prédiction. En effet, la prédiction par ce type de méthode n'est pas directe (Saporta, 2008), par conséquent peut rendre l'arbre instable.

On pourrait considérer les données *Epilepsy* comme étant des données fonctionnelles répétées (Chen and Müller, 2012). En effet, une même quantité (l'accélération) est mesurée suivant plusieurs conditions (axes spatiaux). Ce cas peut être retrouvé dans plusieurs études où les données fonctionnelles multivariées ont pour chaque dimension le même domaine. Il semble nécessaire de fournir des méthodes capables de prendre en compte ces spécificités pour une meilleure interprétation. Le chapitre suivant se concentre sur cette problématique.

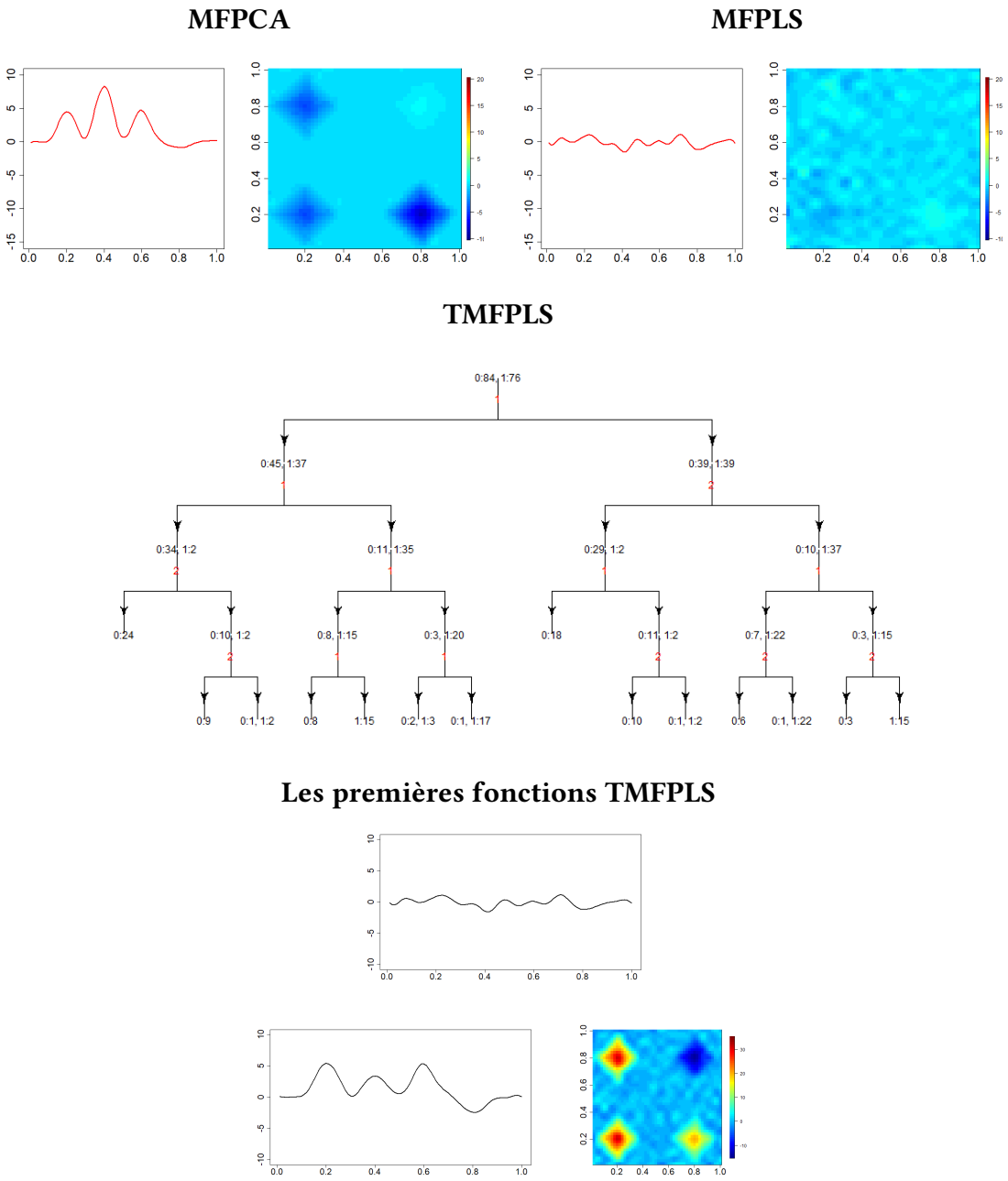

Figure 3.6 – Exemples de modèles estimés avec une seule composante

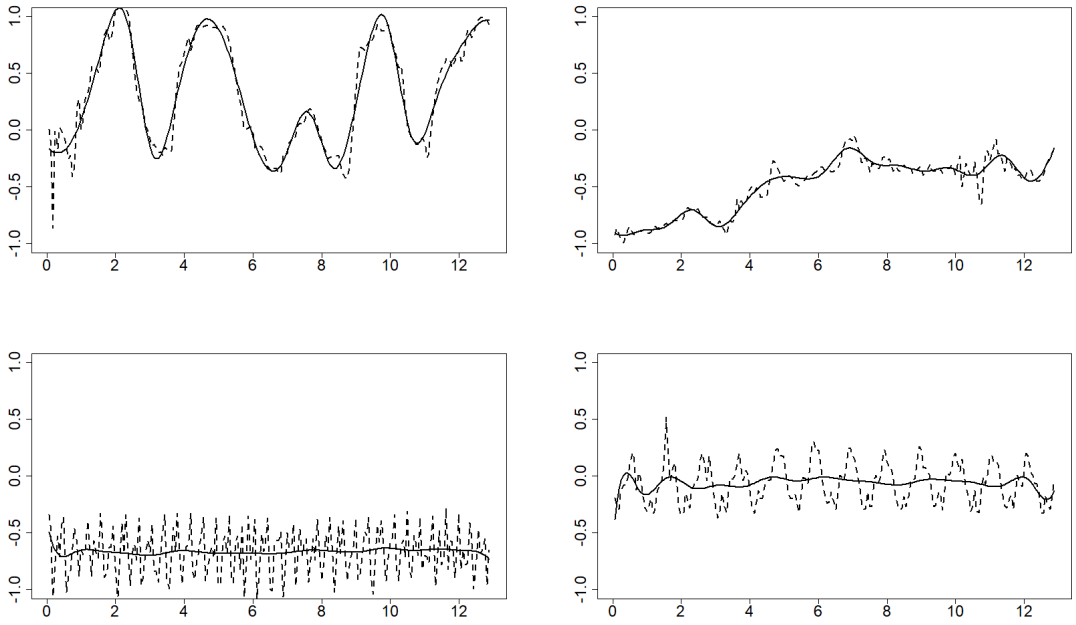

Figure 3.7 – Exemples de lissage sur la base de données *Epilepsy*, en pointillées les fonctions brutes et en gras les fonctions lissées.

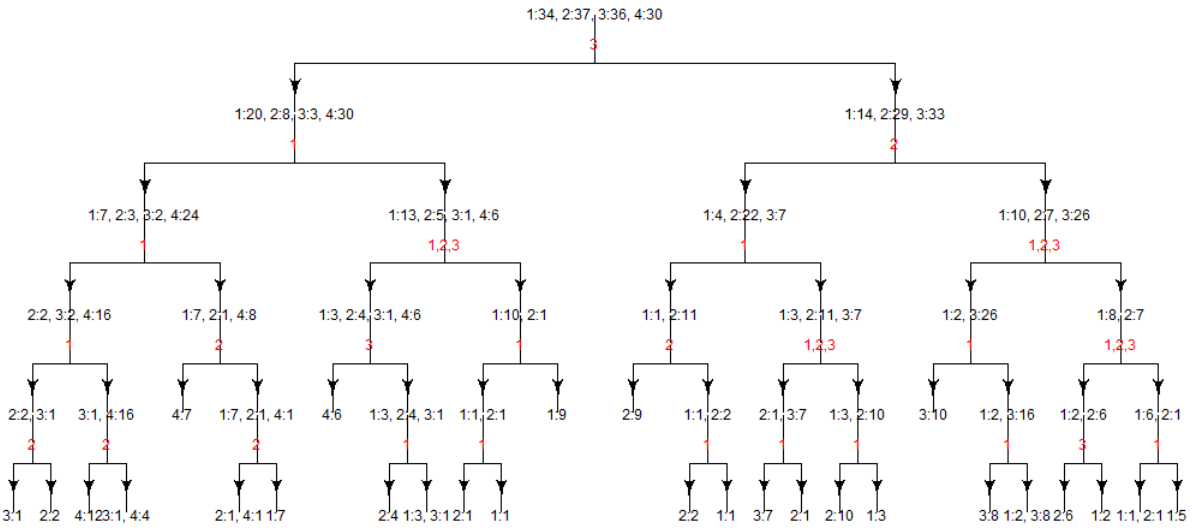

Figure 3.8 – Le modèle estimé TMFPLS

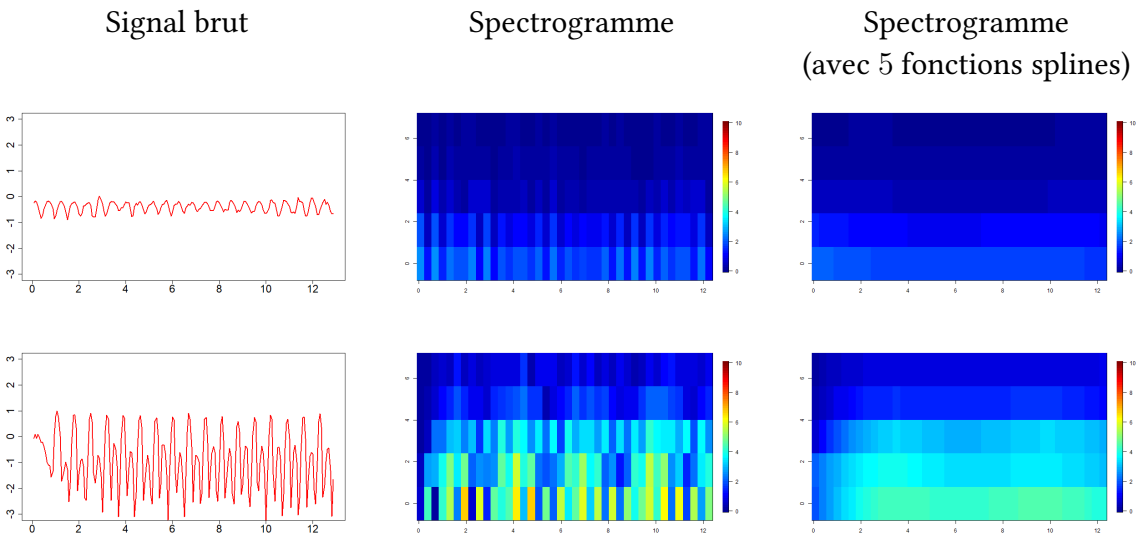

Figure 3.9 – Exemples de spectrogrammes et leurs reconstructions fonctionnelles

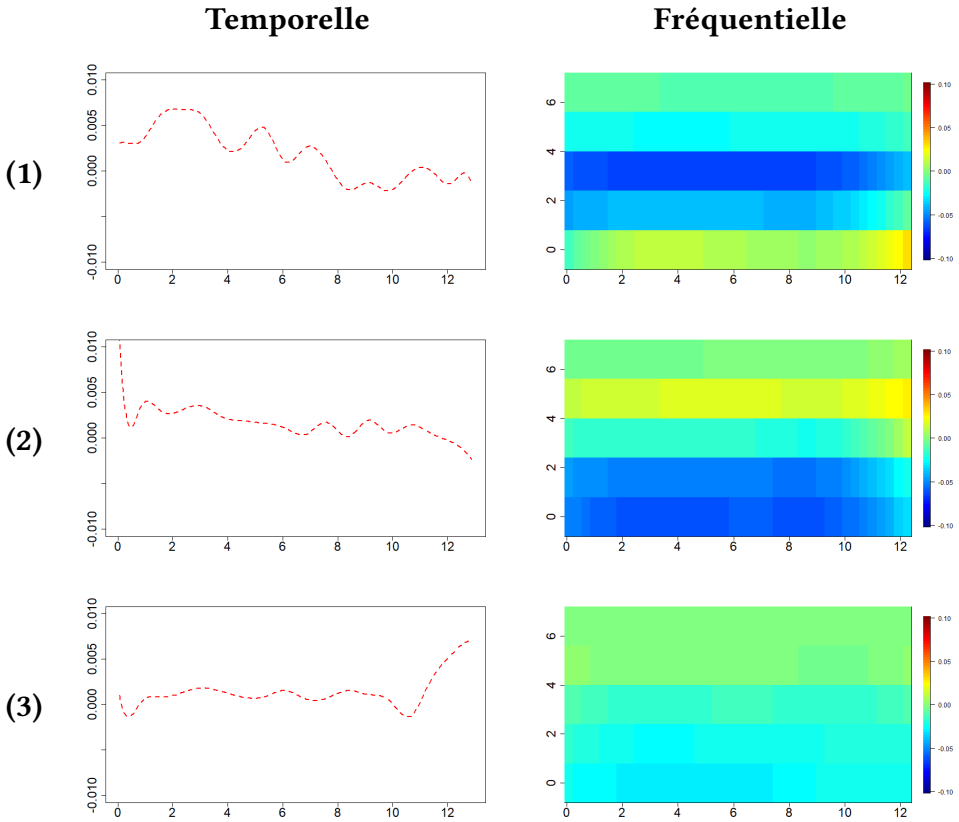

Figure 3.10 – Fonction discriminante par MFPCA.

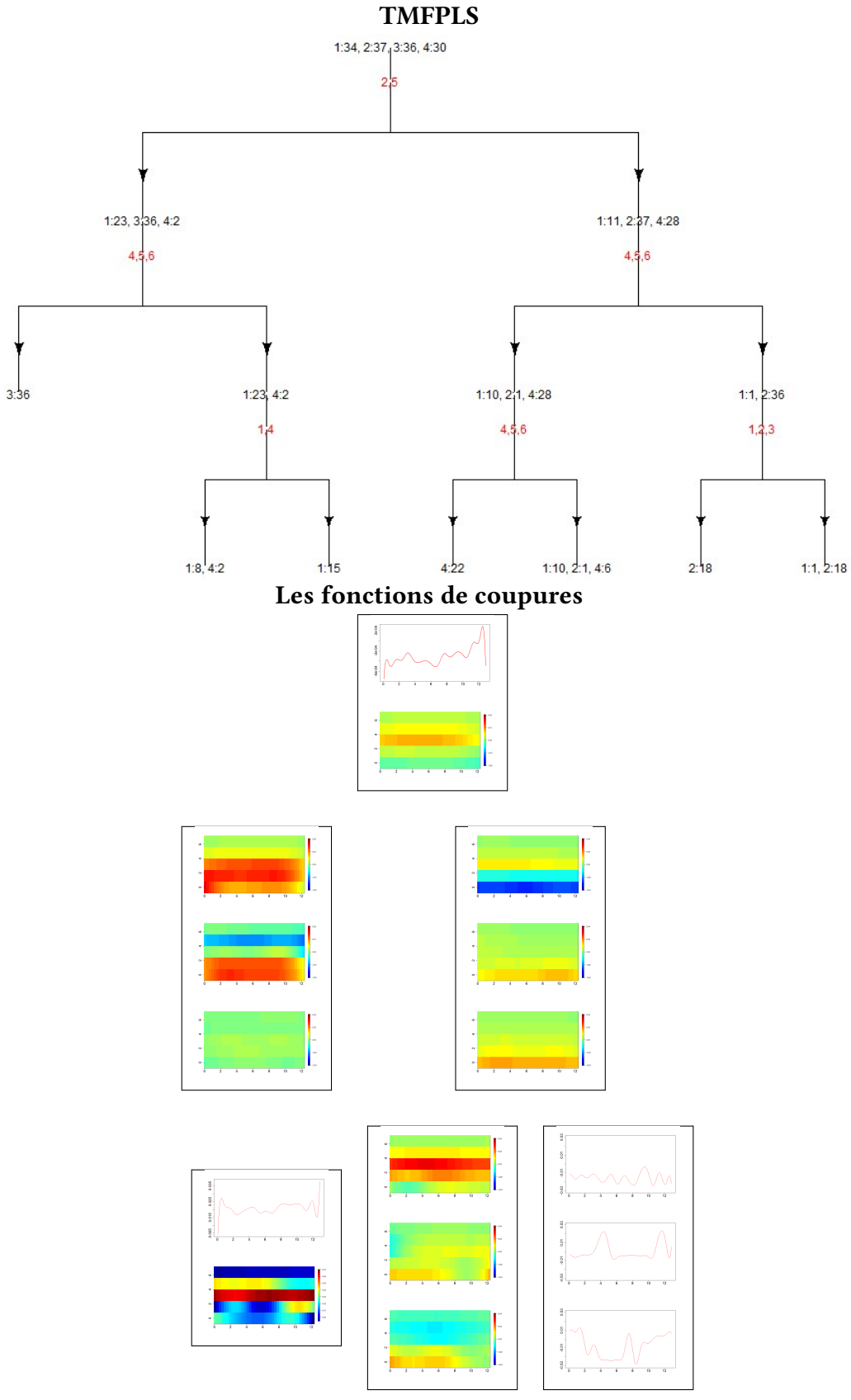

Figure 3.11 – Modèle TMFPLS estimé

Deuxième partie

Données fonctionnelles répétées

Chapitre 4

Modèles de régression avec données fonctionnelles répétées

Ce chapitre traite des modèles de régression linéaire et de classification avec des données fonctionnelles répétées. Pour chaque unité statistique de l'échantillon, une variable à valeur réelle est observée au fil du temps suivant différentes conditions. Deux méthodes pour la régression, basés sur des pénalités de fusion, sont présentés. La première est une généralisation du modèle *variable fusion* fondé sur le graphe 1-NN. La seconde, appelé *group fusion lasso*, suppose une certaine structure de groupes des conditions et favorise l'homogénéité des dimensions de la fonction coefficient au sein de ces groupes. Pour étudier les performances de ces méthodes, nous présentons une étude de simulation et une application sur des données EEG.

4.1 Introduction

Le cadre sur lequel ce chapitre s'intéresse est celui où X, une variable aléatoire fonctionnelle univariée, est observée suivant p différentes conditions : $\mathcal{C}_1, \ldots, \mathcal{C}_p$ avec $p \geq 1$. Ces dernières sont supposées être des éléments de l'espace métrique $(\mathcal{S}, \mathcal{D})$, typiquement $(\mathbb{R}^s, \|\cdot\|_2)$ où s est un entier naturel $s \geq 1$. Autrement dit, $\mathcal{C}_1, \ldots, \mathcal{C}_p$ admettent des relations de proximités induites par la distance $\mathcal{D}$. Par exemple, les conditions peuvent être des instants de temps et/ou des lieux. Nous supposons également que l'espace des fonctions où X prend valeurs est celui des fonctions de carré intégrable $L_2([0,T])$, où $T > 0$ (Ramsey and Silverman, 2005).
Le cadre présenté se prête aux situations où les données sont des enregistrements de plusieurs capteurs censés mesurer la même unité physique. Par exemple, les données d'électroencéphalogramme ou EEG (Ruiz et al., 2021) représentent des mesures de l'activité cérébrale via l'intensité du champ électrique. Elles sont enregistrées en différentes régions du cerveau pour une durée de $T = 500ms$ (voir la Figure 4.1), à l'aide de $p = 28$ électrodes/capteurs répartis uniformément.

Nous désignons par $X^{(j)}$ l'observation de X sous la condition $\mathcal{C}_j$, $j = 1, \ldots, p$, et par $\mathbf{X}$, le vecteur aléatoire suivant :

$$\mathbf{X} = (X^{(1)}, \ldots, X^{(p)})^\top.$$

Les réalisations de $X^{(1)}, \ldots, X^{(p)}$ sont connues sous le nom de données fonctionnelles répétées. Pour ce type de données, une problématique centrale dans la littérature a été la recherche de méthodes d'analyse en composantes principales (ACP) capables de traiter la dépendance entre les composantes $X^{(j)}$. Dans Chen and Müller (2012), les auteurs utilisent une double ACP exploitant la structure métrique de l'espace des conditions $\{\mathcal{C}_1, \ldots, \mathcal{C}_p\}$. Dans Jacques and Preda (2014), cette structure est ignorée et $\mathbf{X}$ est considéré comme un vecteur aléatoire fonctionnel à p dimensions. Les composantes principales sont ensuite utilisées pour la classification non supervisée (*clustering*) et pour la visualisation.

Dans ce chapitre, nous nous intéressons au problème de l'estimation de modèles de régression de $\mathbf{X}$ en Y, où Y est une variable aléatoire binaire ou scalaire. Notre attention se porte plus particulièrement à la prise en compte de la topologie des conditions par le biais de relation de voisinage ou de structure de groupes (définis éventuellement par la distance $\mathcal{D}$) des conditions.

Ainsi, à la différence du cadre classique de la régression linéaire avec des données fonc-

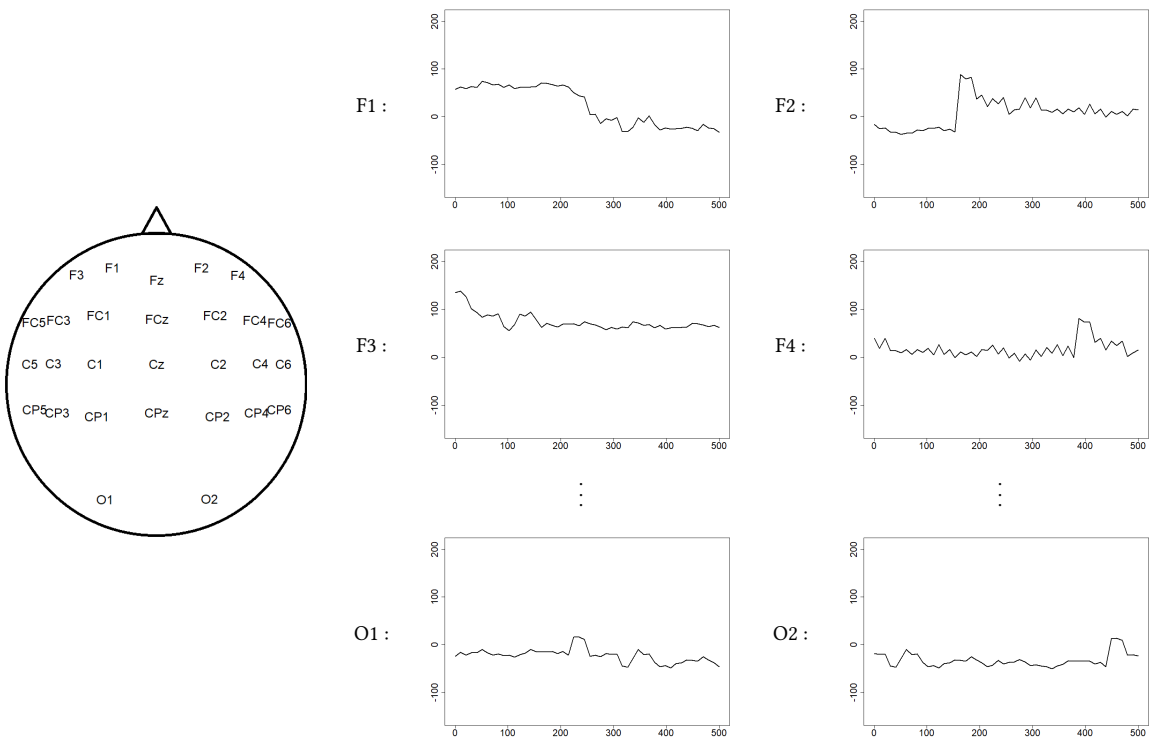

Figure 4.1 – Données *FingerMovements*. Chaque sujet est représenté par $p = 28$ enregistrements EEG (à droite) correspondant à 28 capteurs situés sur *scalp* (à gauche).

tionnelles multivariées, nous utilisons les conditions de mesure des composants $X^{(j)}$ pour l'estimation du modèle. À notre connaissance, aucune méthode proposée dans le cadre des données fonctionnelles multivariées ne prend explicitement en compte les informations apportées par les conditions. Les contributions existantes considèrent principalement $\mathbf{X}$ comme un vecteur fonctionnel à p dimensions et les méthodologies ont été développées pour tenir compte de la dépendance entre les dimensions de $\mathbf{X}$ (voir par exemple Yi et al. (2022), Górecki et al. (2015), Beyaztas and Shang (2022)).

Ici, nous abordons le problème de l'estimation de modèles de régression du point de vue de l'interprétation, dans le sens que les composantes $X^{(j)}$ ayant des conditions proches pourraient fournir des informations similaires dans le modèle de régression. L'application qui motive ce chapitre est la classification de données EEG. Ces données sont issues d'une expérience dont le but est de déterminer dans quelle mesure la capacité d'une personne à être gauchère ou droitière est associée à une activité électrique différente du cerveau. Pour ce faire, p capteurs ont été placés sur le crâne de plusieurs sujets. Ceux-ci sont divisés en deux groupes : les gauchers ($Y = 1$) et les droitiers ($Y = 0$). Il est demandé à chacun de taper un texte, afin de mesurer l'intensité du champ électrique X simultanément à $p = 28$ positions spatiales (capteurs) du *scalp* pendant $T = 500ms$. L'objectif est de prédire Y à l'aide de $\mathbf{X}$.

Le modèle standard de régression linéaire fonctionnelle suppose qu'il existe $\beta^{(0)} \in \mathbb{R}$ et

la fonction coefficient (de régression) $\beta = (\beta^{(1)}, \ldots, \beta^{(p)})^\top \in \mathcal{H} = \{L_2([0,T])\}^p$ tels que

$$\mathbb{E}(Y|\mathbf{X}) \approx \beta^{(0)} + \sum_{j=1}^{p} \int_0^T X^{(j)}(t)\beta^{(j)}(t)dt. \tag{4.1}$$

Si $\{(\mathbf{X}_i, Y_i)\}_{i=1,\ldots,n}$ est un échantillon i.i.d. de taille n, $n \geq 1$, tiré de la même distribution que $(\mathbf{X}, Y)$ et $\{(x_i, y_i)\}_{i=1,\ldots,n}$ des observations de cet échantillon, l'estimation du modèle (4.1) est basée sur la minimisation de la somme des erreurs quadratiques (MSE), c'est-à-dire,

$$(\hat{\beta}_0, \hat{\beta}) = \underset{(\psi_0,\psi)\in\mathbb{R}\times\mathcal{H}}{\arg\min} \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \left(\psi_0 + \sum_{j=1}^{p} \langle x_i^{(j)}, \psi^{(j)} \rangle_{L_2} \right) \right)^2. \tag{4.2}$$

En raison de la non-inversibilité de l'opérateur de covariance, l'estimation directe du coefficient β sous la minimisation du critère MSE est un problème mal posé (Cardot et al., 1999). La régression en composantes principales (PCR) et les moindres carrés partiels (PLS) sont des moyens efficaces pour estimer β dans ce cas (voir par exemple Escabias et al. (2005), Aguilera et al. (2006), Preda and Saporta (2002), Moindjié et al. (2022)). Toutefois, dans ces approches, les fonctions coefficients estimées sont parfois difficiles à interpréter : pourquoi deux composantes $X^{(j)}$ et $X^{(j')}$ qui sont proches dans l'espace des mesures, c'est-à-dire que la distance $\mathcal{D}(\mathcal{C}_j, \mathcal{C}_{j'})$ est faible, ont-elles des fonctions coefficient associées très différentes $\beta^{(j)}$ et $\beta^{(j')}$? Cette situation se produit surtout lorsque p est grand. Dans Godwin (2013), les auteurs proposent d'ajouter la contrainte $\mathcal{P} = \sum_{j=1}^{p} ||\psi^{(j)}||_{L_2}$ dans le modèle de régression. Dans ce cas, $\mathcal{P}$ est une généralisation aux variables fonctionnelles multivariées de la pénalité *group lasso* (GL), introduite à l'origine pour le cas des données multivariées classiques (Meier et al. (2008), Yuan and Lin (2006)).

Cette pénalité permet d'obtenir un compromis entre un nombre minimal de composantes $X^{(j)}$ contribuant au modèle et l'ajustement du modèle aux données d'apprentissage. Notre hypothèse est que la proximité entre les composantes $X^{(j)}$, au sens de la distance $\mathcal{D}$ des conditions correspondantes $\mathcal{C}_j$, peut contribuer à une meilleure interprétation de β. À cette fin, nous nous inspirons de la pénalité *fusion*, introduite dans Land and Friedman (1997) pour le cas multivarié classique.

Soit v une fonction surjective, $v : \{\mathcal{C}_1, \ldots, \mathcal{C}_p\} \mapsto \{1, \ldots, K\}$ et $K \leq p$, la pénalité fusion (dans le cadre fonctionnel) peut être se définit comme suit :

$$\mathcal{P}(\beta) = \sum_{k=1}^{K} \sqrt{\sum_{j\in\mathcal{I}_k} \left|\left|\beta^{(j)} - \bar{\beta}_{\mathcal{I}_k}\right|\right|_{L_2}^2},$$

où pour chaque $k = 1, \ldots, K$, $\mathcal{I}_k = \{j : v(\mathcal{C}_j) = k\}$ et $\bar{\beta}_{\mathcal{I}_k}(t) = \frac{1}{p_k} \sum_{j\in\mathcal{I}_k} \beta^{(j)}(t)$ pour $t \in [0,T]$ avec p_k qui désigne le cardinal de $\mathcal{I}_k$ pour $k = 1, \ldots, p$.

Alors, la proximité entre les conditions $\mathcal{C}_1, \ldots, \mathcal{C}_p$ peut être intégrée par la fonction v et la distance $\mathcal{D} : v^{-1}(k)$ donne les conditions les plus proches de $\mathcal{C}_k$. Cette pénalité favorise les dimensions proches de $\mathbf{X}$ à avoir des coefficients similaires dans la fonction de régression (les fonctions de $\beta^{(j)}$).

À notre connaissance, cette pénalité n'a pas été étudiée dans le cas de la régression avec des variables explicatives fonctionnelles répétées (ni fonctionnelles multivariées). Dans le

cadre multivarié classique, les modèles de régressions linéaires ayant cette pénalité sont connus sous le nom de *variable fusion* (FU) (Land and Friedman, 1997) et, lorsqu'une pénalité lasso est ajoutée, sous le nom de *fused lasso* (Tibshirani et al., 2005). Plus récemment, une extension des conditions uniques aux groupes de conditions de cette pénalité a été introduite sous le nom de *group fused lasso* (GFL), voir Bleakley and Vert (2011). Cependant, dans ces cas, la fonction v a été orientée comme un moyen d'intégrer des conditions (ou dimensions) consécutives, c'est-à-dire que v est définie comme $v(\mathcal{C}_j) = j + 1$, où $1 \leq j \leq p - 1$ et $v(\mathcal{C}_p) = p$. Même si ce cas peut être adapté à la situation où $\mathcal{S} \subset \mathbb{R}$, et les conditions sont des indices consécutifs (par exemple, des instants de temps), en général, cela limite les applications potentielles.

Dans ce chapitre, nous présentons deux nouvelles pénalités de type fusion pour l'estimation de modèle de régression linéaire fonctionnelle. La première (FU) est une extension de la pénalité *variable fusion* pour laquelle le graphe du plus proche voisin (1-NN) est utilisé pour définir v. Nous montrons que l'estimation de FU dans ce cadre est équivalente à l'estimation de modèles de régressions avec une pénalité *group lasso*, tels que ceux étudiés dans Godwin (2013). La seconde pénalité prend en compte des structures de groupe au sein des conditions. Nous l'appelons *group fusion lasso* (GFUL). Elle permet de tester l'égalité entre les dimensions de la fonction coefficient appartenant au même groupe de conditions.

La suite du chapitre est organisée comme suit. La section 4.2 présente les modèles proposés et leurs stratégies d'estimation. Une étude comparative des deux pénalités et de la pénalité *group lasso* est réalisée à l'aide de données simulées dans la section 4.3.1. La section 4.4 présente une application sur données réelles (classification d'EEG) des méthodologies présentées. Finalement, le document se conclut par une discussion à la section 4.5.

4.2 Deux nouvelles pénalités fusion pour la régression linéaire avec des données fonctionnelles multivariées

Sans perte de généralité, nous supposons que $\mathbf{X}$ et Y sont des variables aléatoires de moyenne nulle et que $\{(x_i, y_i)\}_{i=1,\dots,n}$ est la réalisation d'un échantillon aléatoire de taille n, $n \geq 1$, tiré de la distribution conjointe de $(\mathbf{X}, Y)$. Ainsi, le coefficient directeur $\beta^{(0)}$ dans (4.1) est nul et le critère de la moyenne quadratique (4.2) revient à :

$$\hat{\beta} = \underset{\psi \in \mathcal{H}}{\arg\min} \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} \langle x_i^{(j)}, \psi^{(j)} \rangle_{L_2} \right)^2 .$$

On rappelle que pour chaque $i = 1, \dots, n$, $y_i \in \mathbb{R}$ et x_i est une fonction à plusieurs variables définie sur $[0, T]$,

$$x_i(t) = \left(x_i^{(1)}(t), \dots, x_i^{(j)}(t), \dots, x_i^{(p)}(t) \right)^\top, \quad t \in [0, T].$$

La fonction $x_i^{(j)}$ correspond à l'observation de x_i sous la condition $\mathcal{C}_j$, $j = 1, \dots, p$.

Utilisant ces définitions et hypothèses, la partie suivante présente la première pénalité construite à l'aide de la distance entre les conditions.

4.2.1 La méthode *variable fusion* basée sur le graphe 1-NN

L'idée de base de la pénalité présentée ici est que si deux conditions $\mathcal{C}_j$ et $\mathcal{C}_{j'}$ sont proches dans l'espace $\mathcal{S}$ (par rapport à la distance $\mathcal{D}$), alors les contributions apportées par les composantes $X^{(j)}$ et $X^{(j')}$ dans le modèle linéaire (4.1), c'est-à-dire, $\beta^{(j)}$ et $\beta^{(j')}$, pourraient être comparables. Pour cela, nous nous appuyons dans cette partie sur la méthodologie *variable fusion*. En effet, cette dernière, dans le cadre multivarié classique, s'est révélée une bonne candidate pour fournir des modèles parcimonieux capables de rivaliser avec les approches de modèles linéaires existantes (Land and Friedman (1997), Tibshirani et al. (2005), Bleakley and Vert (2011))

Lorsque les conditions $\mathcal{C}_j$ sont des éléments dans $\mathbb{R}^s$ et $s \geq 2$, la distance $\mathcal{D}$ permet de définir des relations de voisinage. Ces dernières peuvent ensuite être utilisées pour estimer β. Plus précisément, en s'inspirant de Land and Friedman (1997) et à l'aide du 1-NN (*1-plus proche voisin*), le modèle de *variable fusion* dans notre cas peut être formulé comme suit :

$$\hat{\beta}_\lambda = \underset{\beta \in \mathcal{H}}{\arg\min} \frac{1}{2} \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} \langle x_i^{(j)}, \beta^{(j)} \rangle_{L_2} \right)^2 + \lambda \sum_{j=1}^{p} ||\beta^{(j)} - \beta^{(v(\mathcal{C}_j))}||_{L_2} \tag{4.3}$$

où $\lambda \geq 0$ est le paramètre de régularisation et $v : \{\mathcal{C}_1, \ldots, \mathcal{C}_p\} \to \{1, \ldots, p\}$ désigne la fonction de voisinage

$$v(\mathcal{C}_j) = \underset{i \in \{1,\ldots,p\} \setminus \{j\}}{\arg\min} \mathcal{D}(\mathcal{C}_i, \mathcal{C}_j), \ j = 1, \ldots, p. \tag{4.4}$$

Ainsi, la fonction v permet d'intégrer dans l'estimation de β les informations apportées par les conditions.

À titre d'exemple, la Figure 4.2 présente une illustration de la fonction v, où $\mathcal{S} \subset \mathbb{R}^2$ et $p = 8$ conditions $\mathcal{C}_j$, $j = 1, \ldots, p$ sont considérées. Dans ce cas, la fonction v donne les correspondances suivantes : $v(\mathcal{C}_1) = 8$, $v(\mathcal{C}_2) = 5$, $v(\mathcal{C}_3) = 4$, $v(\mathcal{C}_4) = 5, v(\mathcal{C}_5) = 4, v(\mathcal{C}_6) = 1, v(\mathcal{C}_7) = 1$ et $v(\mathcal{C}_8) = 1$.

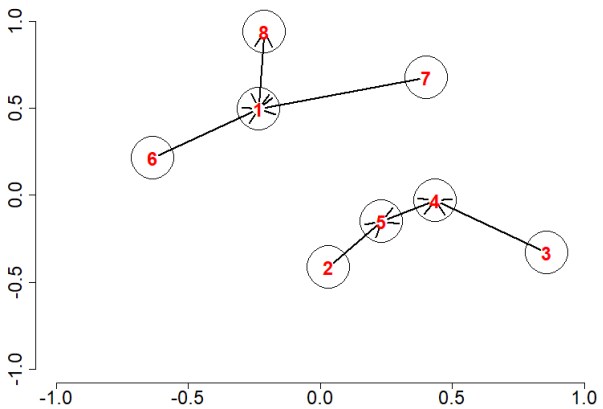

Figure 4.2 – Les 1 plus proches voisins, $a \to b$ indique que b est le voisin de a.

En général, il convient de noter que l'ensemble des $\arg\min$ dans (4.4) n'est pas toujours un singleton. Lorsque cet ensemble est composé de plusieurs éléments, on peut y choisir aléatoirement ou à l'aide d'information a priori un unique voisin.
De plus, on peut remarquer que la fonction de pénalité dans (4.3) peut s'écrire comme

$$\sum_{j=1}^{p} ||\beta^{(j)} - \beta^{(v(\mathcal{C}_j))}||_{L_2} = ||\mathbf{L}\beta||_{L_2,1},$$

où $\mathbf{L} = \mathbf{W} - \mathbb{I}_{p\times p}$, $\mathbf{W} = (w_{i,j})_{1\leq i\leq p, 1\leq j\leq p}$ est la matrice d'adjacence composée des éléments

$$w_{i,j} = \begin{cases} 1 & \text{if } v(\mathcal{C}_i) = j \\ 0 & \text{sinon,} \end{cases}$$

avec $\mathbb{I}_{p\times p}$ la matrice unitaire $p \times p$ et $||\cdot||_{L_2,1}$ définie comme :

$$||f||_{L_2,1} = \sum_{i=1}^{p} ||f^{(i)}||_{L_2},\ f \in \mathcal{H}.$$

Il est facile de voir que le rang de la matrice $\mathbf{L}$ est généralement inférieur à p, puisque des relations symétriques sont possibles (contrairement au cas des conditions consécutives, voir par exemple Land and Friedman (1997)). Par exemple, la Figure 4.2 montre que $\mathcal{C}_1$ est le voisin de $\mathcal{C}_8$ et que $\mathcal{C}_8$ est le voisin de $\mathcal{C}_1$, de même pour le couple $(\mathcal{C}_5, \mathcal{C}_4)$.

Lemme 2. *Soit r est le rang de la matrice $\mathbf{L}$, il existe une matrice $r \times p$ de rang maximal $\mathbf{L}_0$ telle que*

$$||\mathbf{L}f||_{L_2,1} = ||\mathbf{L}_0 f||_{L_2,1},\ f \in \mathcal{H}. \tag{4.5}$$

Ainsi, $\mathbf{L}_0$ permet d'éviter la redondance. Dans notre cas, la construction d'une telle matrice consiste à trouver les deux lignes où se produisent des relations symétriques et à remplacer l'une des lignes par l'autre multipliée par 2. Il convient de noter que r, le rang de la matrice $\mathbf{L}$, coïncide avec le nombre de *vertex* de la version non orientée du graphe 1-NN. À titre d'illustration, dans notre exemple (Figure 4.2), nous avons les matrices suivantes

$$\mathbf{L} = \begin{pmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix}$$

et

$$\mathbf{L}_0 = \begin{pmatrix} -2 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2 & 2 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \end{pmatrix}.$$

Le lemme 2 implique qu'il existe une reformulation alternative de (4.3). La proposition suivante montre que le problème FU peut être mis en relation avec un problème *group lasso*. Pour simplifier la notation, dans cette proposition et dans la suite du chapitre, nous noterons par $\langle , \rangle_{\mathcal{H}}$ le produit scalaire de $f, g \in \mathcal{H}$ dans $\mathcal{H}$

$$\langle f, g \rangle_{\mathcal{H}} = \sum_{i=1}^{p} \langle f^{(i)}, g^{(i)} \rangle_{L_2}.$$

Proposition 3. *La solution du problème* (4.3) *vérifie*

$$\hat{\beta}_\lambda = \mathbf{D}^{-1} \hat{\psi}_\lambda,$$

où

$$\hat{\psi}_\lambda = \underset{f \in \mathcal{H}}{\arg\min} \frac{1}{2} \sum_{i=1}^{n} \left(y_i - \langle (\mathbf{D}^{-1})^\top x_i, f \rangle_{\mathcal{H}} \right)^2 + \lambda \sum_{j=1}^{r} ||f^{(j)}||_{L_2}, \tag{4.6}$$

$\mathbf{D} = \begin{pmatrix} \mathbf{L}_0 \\ \mathbf{T} \end{pmatrix}$, $\mathbf{L}_0$ *est la matrice réduite* $r \times p$ *de* $\mathbf{L}$ *et* $\mathbf{T}$ *est une matrice* $(p-r) \times p$ *dont les lignes forment une base de l'espace nul de* $\mathbf{L}_0$, $\mathbf{L}_0 \mathbf{T}^\top = \mathbf{0}_{r \times (p-r)}$ *et* $\mathbf{0}_{r \times (p-r)}$ *est la matrice* $r \times (p-r)$ *de zéros.*

L'estimation de la partie non pénalisée de f dans (4.6), $f^{(r+1)}, \ldots, f^{(p)}$, peut conduire (en mettant le maximum de poids sur la partie non contrainte de β) à un surajustement du modèle. Pour résoudre ce problème, nous proposons de modifier le terme de pénalité dans (4.6) comme suit :

$$||\mathbf{L}\beta||_{L_2,1} + \frac{\sqrt{p-r}}{\eta} ||\mathbf{T}\beta||_{L_2,2},$$

où η est la norme (matricielle) de Frobenius de $\mathbf{T}$, et $||\cdot||_{L_2,2}$ désigne la norme de Frobenius de $\mathcal{H}$:

$$||f|_{L_2,2} = \sqrt{\sum_{i=1}^{p} ||f^{(i)}||_{L_2}^2}$$

pour $f \in \mathcal{H}$.
Le problème d'optimisation (4.6) devient donc :

$$\hat{\psi}_\lambda = \underset{f \in \mathcal{H}}{\arg\min} \frac{1}{2} \sum_{i=1}^{n} \left(y_i - \langle (\mathbf{D}^{-1})^\top x_i, f \rangle_{\mathcal{H}} \right)^2 + \lambda \left(\sum_{j=1}^{r} ||f^{(j)}||_{L_2} + \frac{\sqrt{p-r}}{\eta} \left(\sum_{j=r+1}^{p} ||f^{(j)}||_{L_2}^2 \right)^{1/2} \right). \tag{4.7}$$

Cette méthode est basée sur un seul voisin. Dans la section suivante, nous présentons une méthode similaire basée sur plus d'un voisin, que nous appelons *group fusion lasso*.

4.2.2 Le modèle *group fusion lasso*

Considérons l'exemple représenté dans la figure 4.3. Dans cet exemple, nous supposons que les conditions sont étiquetées selon $K = 3$ groupes : le groupe jaune $(\mathcal{C}_3, \mathcal{C}_4, \mathcal{C}_7)$, le

groupe rouge $(\mathcal{C}_1, \mathcal{C}_6, \mathcal{C}_8)$ et le groupe bleu$(\mathcal{C}_2, \mathcal{C}_5)$. Pour cette configuration, plus d'un voisin doit être considéré. En effet, les ensembles suivants $(\mathcal{C}_1, \mathcal{C}_6, \mathcal{C}_8)$ $(\mathcal{C}_3, \mathcal{C}_4, \mathcal{C}_7)$, $(\mathcal{C}_2, \mathcal{C}_5)$ ont des relations de voisinage symétriques (c'est-à-dire que $\mathcal{C}_1$ a pour voisins $(\mathcal{C}_8, \mathcal{C}_6)$, $\mathcal{C}_6$ a pour voisins $(\mathcal{C}_1, \mathcal{C}_8)$, etc.). Plutôt que d'examiner les interactions des conditions individuellement, ce qui complexifie l'estimation du modèle (Tibshirani et al., 2005), nous proposons dans cette section de tester les relations de groupes qui en résultent.

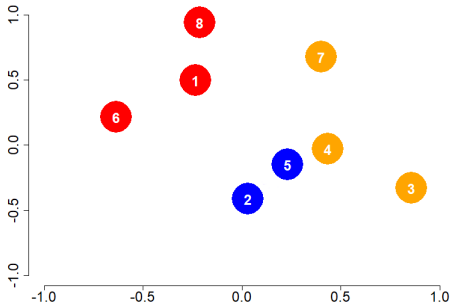

Figure 4.3 – Conditions organisées en groupes

La structure de groupe des conditions est donnée par la fonction v,

$$v : \{\mathcal{C}_1, \ldots, \mathcal{C}_p\} \to \{1, \ldots, K\}, \tag{4.8}$$

où K est le nombre de groupes, $K \leq p$.
On rappelle la définition des ensembles $\mathcal{I}_k$

$$\mathcal{I}_k = \{j \in \{1, \ldots, p\},\ v(\mathcal{C}_j) = k\},\ k = 1, \ldots, K.$$

Nous noterons par p_k l'effectif (cardinal) de chaque groupe $\mathcal{I}_k$

$$p_k = |\mathcal{I}_k|,\ k = 1, \ldots, K.$$

L'idée derrière la méthodologie *group fusion* (GFU) est d'introduire un critère favorisant l'obtention de coefficients similaires pour les composantes correspondant à des conditions appartenant au même groupe. Dans l'exemple présenté dans la figure 4.3, $p = 8$, $K = 3$ et

- $v(\mathcal{C}_1) = v(\mathcal{C}_6) = v(\mathcal{C}_8) = 1$, le groupe "rouge",
- $v(\mathcal{C}_2) = v(\mathcal{C}_5) = 2$, le groupe "bleu",
- $v(\mathcal{C}_3) = v(\mathcal{C}_4) = v(\mathcal{C}_7) = 3$ le groupe "jaune".

Ainsi, comme dans le cadre de la régularisation lasso, cette méthodologie d'estimation oblige les groupes de conditions à avoir des fonctions coefficient proches et, éventuellement, à ce que certaines d'entre elles soient exactement les mêmes :

$$\{\beta^{(1)} = \beta^{(6)} = \beta^{(8)}\} \text{ et/ou} \{\beta^{(2)} = \beta^{(5)}\} \text{et/ou } \{\beta^{(3)} = \beta^{(4)} = \beta^{(7)}\}.$$

À cette fin, modifions le critère (4.3) en ajoutant une pénalité pour chaque groupe k de fonctions coefficient, $\mathcal{P}_k(\cdot)$, $k = 1, \ldots, K$, comme suit :

$$\begin{aligned} \hat{\beta}_\lambda &= \arg\min_{\beta \in \mathcal{H}} \frac{1}{2} \sum_{i=1}^{n} (y_i - \langle x_i, \beta \rangle_{\mathcal{H}})^2 + \lambda \sum_{k=1}^{K} \mathcal{P}_k(\beta) \\ &\text{avec } \mathcal{P}_k(\beta) = \sqrt{p_k} \sqrt{\sum_{i \in \mathcal{I}_k} ||\beta^{(i)} - \bar{\beta}_{\mathcal{I}_k}||_{L_2}^2} \end{aligned} \tag{4.9}$$

et $\bar{\beta}_{\mathcal{I}_k}(t) = \frac{1}{p_k} \sum_{j \in \mathcal{I}_k} \beta^{(j)}(t)$, $t \in [0, T]$.

Remarque 6. *Si, pour un certain $k \in \{1, \ldots, K\}$, $\mathcal{I}_k = \{j\}$, alors il n'y a pas de pénalité sur la j-ième composante (dimension) de la fonction coefficient correspondante, $\beta^{(j)}$.*

Comme pour le critère précédent (4.3), le critère d'optimisation (4.9) peut conduire à un surajustement du modèle (voir la proposition 4) : les pénalités de fusion n'ont aucun contrôle sur tous les termes de la norme de β. Pour surmonter cette difficulté, nous introduisons la méthodologie *group fusion lasso* (GFUL) en s'inspirant de la stratégie elastic-net (Zou and Hastie, 2005), c'est-à-dire,

$$\begin{aligned} \hat{\beta}_{\lambda,\alpha} &= \arg\min_{\beta \in \mathcal{H}} \frac{1}{2} \sum_{i=1}^{n} (y_i - \langle x_i, \beta \rangle_{\mathcal{H}})^2 + \lambda \sum_{k=1}^{K} \mathcal{P}_{\alpha,k}(\beta) \\ &\text{avec } \mathcal{P}_{\alpha,k}(\beta) = (1-\alpha)\mathcal{P}_k(\beta) + \alpha ||\bar{\beta}_{\mathcal{I}_k}||_{L_2}, \quad \alpha \in]0, 1[. \end{aligned} \tag{4.10}$$

L'objectif de GFUL est proche de la méthodologie *group lasso* où, étant donné une certaine structure de groupe des variables explicatives, l'objectif est de réduire à zéro tous les coefficients associés aux variables au sein de certains groupes (pour plus de détails, voir Meier et al. (2008), Yuan and Lin (2006)). De ce point de vue, GFUL vise à obtenir, pour certains groupes, des coefficients de mêmes valeurs ; ce qui offre un cadre plus général.

Remarque 7. *La fonction de pénalité est composée de deux termes : le premier, $\mathcal{P}_k(\beta)$ est de type fusion ; $\mathcal{P}_k(\beta)$ est nul si et seulement si $\beta^{(j)} = \beta^{(k)}$, $\forall j, k \in \mathcal{I}_k$; le second terme, $||\bar{\beta}_{\mathcal{I}_k}||_{L_2}$ est une pénalité de type group-lasso.*

Comme pour la méthodologie FU (voir la proposition 3), nous montrons maintenant que l'estimation GFUL se réduit à une estimation de *group lasso*.
D'abord, notons que dans la méthodologie GFUL, l'appartenance des conditions à des groupes est une notion centrale. Nous définissons la matrice $\mathbf{M}$, dite d'appartenance, comme une matrice permettant d'encoder la structure induite par $\{\mathcal{I}_k\}_{k=1}^K$:

$$\mathbf{M} = \{m_{k,j}\}_{1 \leq k \leq K, 1 \leq j \leq p} \text{ et } m_{k,j} = \begin{cases} 1 & \text{si } j \in \mathcal{I}_k \\ 0 & \text{sinon.} \end{cases}$$

Pour fixer les idées, dans l'exemple d'illustration (Figure 4.3), la matrice d'appartenance est donnée par

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \end{pmatrix}.$$

En général, à une permutation de colonnes près, la matrice $\mathbf{M}$ peut s'écrire

$$\mathbf{M} = \begin{pmatrix} \vec{1}_{p_1}^\top & \vec{0}_{p_2}^\top & \dots & 0 \\ \vec{0}_{p_1}^\top & \vec{1}_{p_2}^\top & \dots & 0 \\ \dots & & & \dots \\ \vec{0}_{p_1}^\top & \vec{0}_{p_2}^\top & \dots & \vec{1}_{p_K}^\top \end{pmatrix},$$

où $\vec{1}_{p_k}$, $\vec{0}_{p_k}$ sont respectivement les vecteurs p_k-colonne de 1 et de 0.
Nous noterons par $\bar{\mathbf{M}}$ la version standardisée de $\mathbf{M}$, i.e $\bar{\mathbf{M}} = \text{diag}(1/p_1, 1/p_2, \dots, 1/p_K)\mathbf{M}$.
Alors, de la même manière que dans le Lemme 2, le résultat suivant s'applique :

Lemme 3. *Soit $f \in \mathcal{H}$, $\alpha \in]0,1[$ et $p_k \geq 2$, pour $k = 1, \dots, K$. Considérons les $2K$ groupes synthétiques $\{\tilde{\mathcal{I}}_k\}_{k=1}^{2K}$, suivants*

$$\tilde{\mathcal{I}}_k = \begin{cases} \left\{ j \in \{1, \dots, p\} \,\middle|\, 1 + \sum_{l=1}^{k-1}(p_l - 1) \leq j \leq \sum_{l=1}^{k}(p_l - 1) \right\} & k = 1, \dots, K \\ \{k + p - 2K\} & k = K+1, \dots, 2K. \end{cases}$$

La pénalité GFUL peut s'écrire comme

$$\sum_{k=1}^{K} \mathcal{P}_{\alpha,k}(f) = \sum_{k=1}^{2K} \sqrt{\sum_{i \in \tilde{\mathcal{I}}_k} ||(\mathbf{G}_\alpha f)^{(i)}||_{L_2}^2}, \; f \in \mathcal{H} \tag{4.11}$$

où $\mathbf{G}_\alpha$ est la matrice inversible $p \times p$, donnée (à une permutation de colonnes près) par :

$$\mathbf{G}_\alpha = \begin{pmatrix} (1-\alpha)\mathbf{R} \\ \alpha \bar{\mathbf{M}} \end{pmatrix},$$

avec $\mathbf{R}$ est la matrice bloc diagonale composée des éléments suivants $\sqrt{p_1}\mathrm{R}_1, \dots, \sqrt{p_K}\mathrm{R}_K$, et pour $k = 1, \dots, K$, R_k est la matrice triangulaire supérieure $(p_k - 1) \times p_k$ matrice obtenues par la décomposition QR de $\mathrm{P}_k = \mathbb{I}_{p_k \times p_k} - \frac{1}{p_k}\mathbf{1}_{p_k \times p_k}$; ici $\mathbf{1}_{p_k \times p_k}$ représente la matrice $p_k \times p_k$ de un.

En utilisant la non-singularité de $\mathbf{G}_\alpha$, la proposition suivante fournit un moyen de résoudre (4.10) à l'aide d'un problème plus simple.

Proposition 4. *La solution de* (4.10)*, vérifie*

$$\hat{\beta}_{\alpha,\lambda} = \mathbf{G}_\alpha^{-1} \hat{\psi}_\lambda,$$

où

$$\hat{\psi}_\lambda = \underset{f \in \mathcal{H}}{\arg\min} \frac{1}{2} \sum_{i=1}^{n} (y_i - \langle (\mathbf{G}_\alpha^{-1})^\top x_i, f \rangle_{\mathcal{H}})^2 + \lambda \sum_{k=1}^{2K} \sqrt{\sum_{i \in \tilde{\mathcal{I}}_k} ||f^{(j)}||_{L_2}^2}. \tag{4.12}$$

Remarque 8. *Le cas où $\alpha = 0$ ou $\alpha = 1$ peut être résolu en utilisant la même technique que dans la Proposition 3. En effet, dans ces cas, la partie non nulle de $\mathbf{G}_\alpha$ est de rang maximum.*

L'estimation directe de β est généralement un problème inverse mal posé (Cardot et al. (1999), Aguilera et al. (2006)). L'expansion en base de fonctions est une technique de réduction de dimension bien connue pour contourner cette difficulté. Elle est présentée dans la section suivante.

4.2.3 Estimation par l'expansion en base de fonctions

La technique d'expansion de base suppose qu'il existe un ensemble de fonctions linéairement indépendantes $\{\phi_k\}_{k=1}^{M}$, de sorte que, pour $i = 1, \ldots, n$, x_i puisse s'écrire sous la forme suivante

$$x_i^{(j)}(t) = \sum_{k=1}^{M} a_{i,k}^{(j)} \phi_k(t) = (a_i^{(j)})^\top \phi(t), \ t \in [0, T] \tag{4.13}$$

où $a_{i,k}^{(j)} \in \mathbb{R}$ pour $i = 1, \ldots, n$, $j = 1, \ldots, p$ et

- $a_i^{(j)} = \begin{pmatrix} a_{i,1}^{(j)} & \ldots & a_{i,M}^{(j)} \end{pmatrix}^\top$,
- $\phi = \begin{pmatrix} \phi_1 & \ldots & \phi_M \end{pmatrix}^\top$ est un vecteur de fonctions.

Alors, pour tout x_i, on a

$$x_i = \begin{pmatrix} x_i^{(1)} \\ \vdots \\ x_i^{(p)} \end{pmatrix} = \Phi a_i$$

où

$$a_i = \begin{pmatrix} a_i^{(1)} \\ \vdots \\ a_i^{(p)} \end{pmatrix} \text{ et } \Phi = \begin{pmatrix} \phi_1 & \ldots & \phi_M & 0 & \ldots & 0 & \ldots & 0 & \ldots & 0 \\ 0 & \ldots & 0 & \phi_1 & \ldots & \phi_M & \ldots & 0 & \ldots & 0 \\ \vdots & & & & & & & & & \vdots \\ 0 & \ldots & 0 & 0 & \ldots & 0 & \ldots & \phi_1 & \ldots & \phi_M \end{pmatrix}.$$

Il est important de noter que dans l'expression de (4.13), nous utilisons la même base ϕ pour toutes les dimensions de $\mathbf{X}$. Cela semble réaliste puisque $X^{(1)}, \ldots, X^{(p)}$ mesurent le même paramètre X. Cependant, cela n'est pas obligatoire. Chaque dimension $X^{(j)}$ peut être exprimée sur une base de fonctions différente.
De même que pour x_i, nous supposons que la fonction coefficient β peut également être exprimée comme suit :

$$\beta(t) = \Phi(t) b, \ t \in [0, T]$$

où

$$b = \begin{pmatrix} b^{(1)} \\ \vdots \\ b^{(p)} \end{pmatrix}, \text{ avec } b^{(j)} \in \mathbb{R}^M.$$

Notons que x_i et β admettent aussi les expressions matricielles suivantes :

$$\beta(t) = \mathbf{B}\phi(t) \text{ et } x_i(t) = \mathbf{A}_i \phi(t) \tag{4.14}$$

où $\mathbf{A}_i$ and $\mathbf{B}$ sont des matrices $p \times M$,
$\mathbf{B} = \begin{pmatrix} b^{(1)} & \ldots & b^{(p)} \end{pmatrix}^\top$ et $\mathbf{A}_i = \begin{pmatrix} a_i^{(1)} & a_i^{(2)} & \ldots & a_i^{(p)} \end{pmatrix}^\top$, pour $i = 1, \ldots, n$.

Proposition 5.

1. $||\beta||_{L_2,1} \doteq \sum_{j=1}^{p} ||\beta^{(j)}||_{L_2} = ||\mathbf{B}\mathrm{F}^{1/2}||_{2,1}$, *où* $||\cdot||_{2,1}$ *est la norme matricielle* $(2,1)$ *et* $\mathrm{F}^{1/2}$ *est la matrice racine carrée de* $\mathrm{F} = \{\langle \phi_i, \phi_j \rangle\}_{i,j}$.
2. *Soit* k *un entier dans* $\{0, \ldots, p-1\}$ *et* $\mathbf{Z}$ *une matrice* $(p-k) \times p$. *Si* $\beta_0(t) = \mathbf{Z}\beta(t)$, *pour* $t \in [0,T]$, *alors*
$$\beta_0(t) = b_0 \Phi(t), \quad \forall t \in [0,T],$$
où $b_0 = (\mathbf{Z} \otimes \mathbb{I}_{M\times M})b$ *et* $\otimes$ *est le produit de Kronecker.*

Le premier point de cette proposition indique que la norme de β dépend du vecteur b et de la base $\{\phi_k\}_{k=1}^{M}$ via la matrice F.
À titre d'exemple, considérons le problème de *group lasso* suivant (chaque dimension représente un groupe)

$$\hat{\beta}_\lambda = \underset{\beta \in \mathcal{H}}{\arg\min} \frac{1}{2} \sum_{i=1}^{n} \left(y_i - \langle x_i, \beta \rangle_{\mathcal{H}}\right)^2 + \lambda \sum_{j=1}^{p} ||\beta^{(j)}||_{L_2} \tag{4.15}$$

Comme $\langle x_i^{(j)}, \beta^{(j)} \rangle = (a_i^{(j)})^\top \mathrm{F} b^{(j)}$ et $||\beta^{(j)}||_{L_2} = \left((b^{(j)})^\top \mathrm{F} b^{(j)}\right)^{\frac{1}{2}}$, le problème (4.15) est équivalent à celui consistant à trouver le vecteur

$$\hat{b}_\lambda = \begin{pmatrix} \hat{b}_\lambda^{(1)} \\ \hat{b}_\lambda^{(2)} \\ \vdots \\ \hat{b}_\lambda^{(p)} \end{pmatrix}$$

tel que

$$\hat{b}_\lambda = \underset{b \in \mathbb{R}^{pM}}{\arg\min} \frac{1}{2} \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} (a_i^{(j)})^\top \mathrm{F} b^{(j)} \right) + \lambda \sum_{j=1}^{p} ||\mathrm{F}^{1/2} b^{(j)}||_2. \tag{4.16}$$

Soit $\hat{\gamma}^{(j)} = \mathrm{F}^{1/2} b^{(j)}$, alors, trouver $\hat{\gamma}_\lambda$, la solution de

$$\hat{\gamma}_\lambda = \underset{\gamma \in \mathbb{R}^{pM}}{\arg\min} \frac{1}{2} \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} (a_i^{(j)})^\top \mathrm{F}^{1/2} \gamma^{(j)} \right) + \lambda \sum_{j=1}^{p} ||\gamma^{(j)}||_2, \tag{4.17}$$

permet d'estimer $b^{(j)}$ comme

$$\hat{b}_\lambda^{(j)} = (\mathrm{F}^{1/2})^{-1} \hat{\gamma}_\lambda^{(j)}, \quad j = 1, \ldots, p. \tag{4.18}$$

Le problème (4.15) est étudié dans Godwin (2013) à l'aide de l'analyse en composante principale afin d'éviter les problèmes de multicolinéarités et de grande dimension(Aguilera et al. (2006), Escabias et al. (2005)).

Le deuxième énoncé de la proposition 5 montre la relation entre les coefficients d'expansion d'une fonction dans $\mathcal{H}$ (en particulier β) dans la base ϕ avec sa transformée linéaire. Dans la section suivante, cette relation permet d'estimer la fonction coefficient dans le cadre de la méthodologie FU en le réduisant à un problème *group lasso*, similaire à (4.19).

Estimation de FU

Afin d'obtenir la solution $\hat{\beta}_\lambda$ du critère FU (4.7), nous utilisons le deuxième énoncé de la Proposition 5 et de la Proposition 3. Soit $\hat{\gamma}_\lambda$ la solution du problème de minimisation

$$\hat{\gamma}_\lambda = \underset{\gamma \in \mathbb{R}^{pM}}{\arg\min} \frac{1}{2} \sum_{i=1}^{n} \left(y_i - a_i^\top (\mathbf{D} \otimes \mathbb{I}_{M\times M})^{-1} \mathbf{F}^{1/2} \gamma \right)^2 + \lambda \left(\sum_{j=1}^{r} ||\gamma^{(j)}||_2 + \frac{\sqrt{p-r}}{\eta} \sqrt{\sum_{j=r+1}^{p} ||\gamma^{(j)}||_2^2} \right),$$

où $\mathbf{F}^{1/2} = \begin{pmatrix} \mathrm{F}^{1/2} & 0 & \dots & 0 \\ 0 & \mathrm{F}^{1/2} & \dots & 0 \\ \vdots & & & \vdots \\ 0 & 0 & \dots & \mathrm{F}^{1/2} \end{pmatrix}$.

Alors, la fonction coefficient $\hat{\beta}_\lambda$ est donnée par

$$\hat{\beta}_\lambda = \Phi(\mathbf{D} \otimes \mathbb{I}_{M\times M})(\mathbf{F}^{1/2})^{-1} \hat{\gamma}_\lambda.$$

Estimation de GFUL

Nous utilisons une procédure similaire à celle de la section précédente pour l'estimation de $\hat{\beta}_{\lambda,\alpha}$.

Soient les ensembles $\mathcal{G}_1, \dots, \mathcal{G}_{2K}$,

$$\mathcal{G}_k = \left\{ j \,\middle|\, 1 + M \sum_{l=1}^{k-1} (p_k - 1) \le j \le M \sum_{l=1}^{k} (p_k - 1) \right\} \qquad k = 1, \dots, K,$$

$$\mathcal{G}_k = \{ j \,|p' + M(k - 1 - K) + 1 \le j \le p' + M(k - K) \} \qquad k = K+1, \dots, 2K,$$

où $p' = M(p-K)$ et K est le nombre de groupes dans GFUL. Les groupes $\{\mathcal{G}_k\}_{k=1}^{2K}$ correspondent à $\{\tilde{\mathcal{I}}_k\}_{k=1}^{2K}$ (voir le lemme 3) sous l'hypothèse de l'expansion en base de fonctions, c.-à-d. quand chaque $\beta^{(j)}$ est représenté par M coefficients d'expansion. Pour faciliter la notation, définissons la matrice de permutation $\mathbf{S} = (s_{u,v} \in \{0,1\})_{(u,v)\in\{1,\dots,p\}^2}$, telle que

$$\mathbf{S}\beta = \begin{pmatrix} \beta_{\mathcal{I}_1} \\ \beta_{\mathcal{I}_2} \\ \dots \\ \beta_{\mathcal{I}_K} \end{pmatrix},$$

où $\beta_{\mathcal{I}_k}$ est le vecteur des composantes de β correspondant à l'ensemble des indices $\mathcal{I}_k$, $k = 1, \dots, K$.

Le problème du *group fusion lasso* se réduit alors à déterminer $\hat{\gamma}_{\lambda,\alpha}$, la solution du problème

$$\hat{\gamma}_{\lambda,\alpha} = \underset{\gamma \in \mathbb{R}^{pM}}{\arg\min} \frac{1}{2} \sum_{i=1}^{n} \left(y_i - a_i^\top (\mathbf{G}_\alpha \mathbf{S} \otimes \mathbb{I}_{M\times M})^{-1} \mathbf{F}^{1/2} \gamma \right)^2 + \lambda \sum_{k=1}^{2K} ||\gamma_{\mathcal{G}_k}||_2. \tag{4.19}$$

En effet, $\hat{\beta}_{\lambda,\alpha}$ vérifie

$$\hat{\beta}_{\lambda,\alpha} = \Phi(\mathbf{G}_\alpha \mathbf{S} \otimes \mathbb{I}_{M\times M})(\mathbf{F}^{1/2})^{-1} \hat{\gamma}_{\lambda,\alpha}.$$

4.2.4 FU et GFUL pour le modèle de régression logistique

Le cas d'une réponse binaire peut être naturellement pris en compte dans les méthodes que nous proposons. Plus précisément, comme dans Meier et al. (2008), le critère MSE est remplacé par celui de la vraisemblance (multiplié par -1) tandis que les termes pénalisés sont les mêmes. Par exemple, dans ce cas, le problème d'optimisation dans (4.3) devient :

$$\hat{\beta}_\lambda = \arg\min_{\beta\in\mathcal{H}} -\sum_{i=1}^{n} (y_i\langle x_i,\beta\rangle_{\mathcal{H}} - \log(1+\langle x_i,\beta\rangle_{\mathcal{H}})) + \lambda\sum_{j=1}^{p} ||\beta^{(j)} - \beta^{(v(\mathcal{C}_j))}||_2. \quad (4.20)$$

4.3 Étude de simulations

Nous présentons une étude de simulation qui vise à comparer les performances des méthodes proposées, FU et GFUL, avec les méthodes lasso concurrentes. Il convient de noter que toutes nos méthodes sont estimées à l'aide du package Lukas and Meier (2020). Les codes R de nos simulations sont disponibles à l'adresse suivante : https://github.com/imoindjie/GFUL-FU.

4.3.1 La configuration de la simulation

Le cadre de la simulation est le suivant. Afin de montrer l'efficacité de la prise en compte de la structure de groupe des conditions, nous considérons deux scénarios. Dans le premier, le nombre de conditions est fixé à $p = 12$ et nous montrons que toutes les méthodes obtiennent des performances similaires en termes de critères MSE. Dans le second, nous augmentons le nombre de conditions à $p = 80$ et nous montrons alors l'efficacité de notre méthode par rapport aux autres. Dans les deux scénarios, le nombre de groupes est $K = 4$ et le nombre de conditions dans chaque groupe est $p_1 = p_2 = \ldots = p_K = \dfrac{p}{K} \doteq \kappa$.
Notre étude se scinde en les parties suivantes.

(a) les conditions et la structure de groupe,

(b) les fonctions coefficients,

(c) les variables explicatives et la variable à expliquer,

(d) Les deux scénarios de simulation,

(e) les méthodes concurrentes,

(f) la qualité de l'ajustement et autres mesures.

(a) Les conditions et la structure de groupe.
Considérons les p conditions $\mathcal{C}_j$, $j = 1,\ldots,p$, comme des points dans $\mathbb{R}^2$ et leur structure

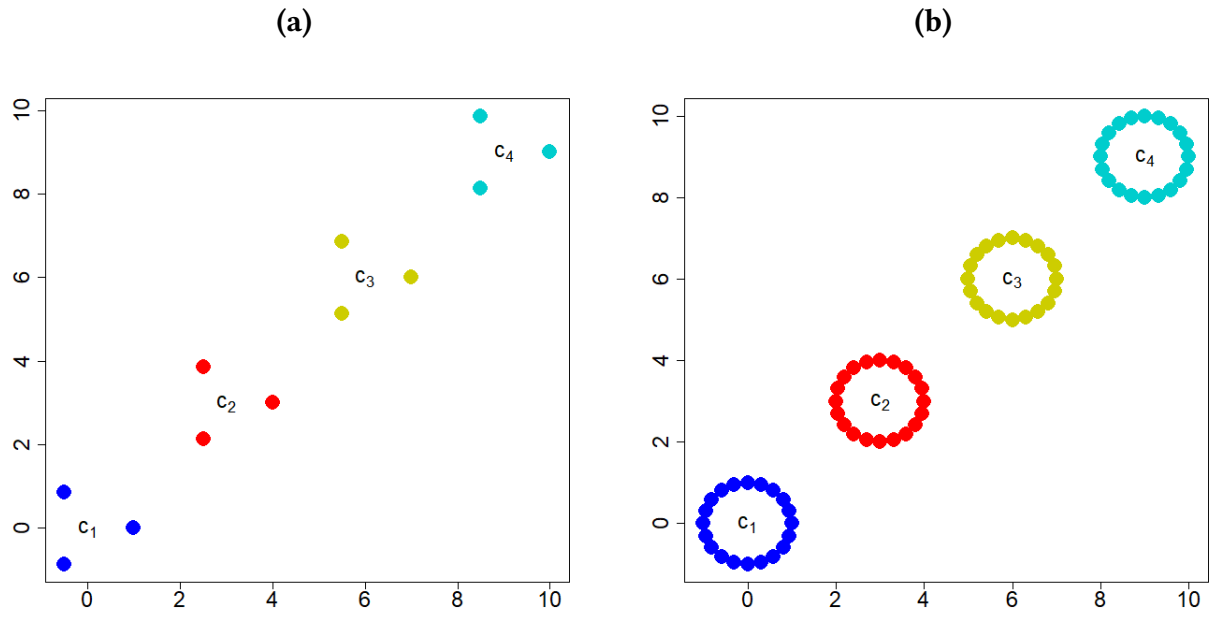

FIGURE 4.4 – Conditions lorsque $p = 12$ **(a)** et $p = 80$ **(b)**. Les couleurs sont associées à chaque groupe de conditions.

de groupe définie comme suit :

$$\begin{aligned}
&\text{Group 1}: \ \mathcal{C}_j = \zeta_j + c_1, && j = 1, \ldots, \kappa, \\
&\text{Group 2}: \ \mathcal{C}_j = \zeta_j + c_2, && j = \kappa + 1, \ldots, 2\kappa, \\
&\text{Group 3}: \ \mathcal{C}_j = \zeta_j + c_3, && j = 2\kappa + 1, \ldots, 3\kappa, \\
&\text{Group 4}: \ \mathcal{C}_j = \zeta_j + c_4, && j = 3\kappa + 1, \ldots, p,
\end{aligned}$$

où

$$\zeta_j = \left(\cos(2\pi \frac{j \bmod \kappa}{\kappa}), \sin(2\pi \frac{j \bmod \kappa}{\kappa}) \right)^\top,$$

et $c_1 = (0,0)^\top$, $c_2 = (3,3)^\top$, $c_3 = 2c_2$, $c_4 = 3c_2$ sont les "centres" des groupes.

La Figure 4.4 présente les conditions pour $p = 12$ et $p = 80$. On peut imaginer qu'elles correspondent à la position de p points sur une pièce de métal carrée 10×10 où l'on observe en chaque point $\mathcal{C}_j, j = 1, \ldots, p$, la température $X^{(j)}$ sur l'intervalle de temps $[0, 1]$.

(b) Les fonctions coefficients.
La fonction coefficient $\beta = (\beta^{(1)}, \ldots, \beta^{(p)})^\top$ est définie comme suit :

$$\begin{aligned}
&\text{Groupe 1 :} && \beta^{(j)} = 0, && j = 1, \ldots, \kappa,\\
&\text{Groupe 2 :} && \beta^{(j)} = \sqrt{2}\sum_{k=1}^{3}\Delta_k, && j = \kappa+1, \ldots, 2\kappa,\\
&\text{Groupe 3 :} && \beta^{(j)} = b_j\sum_{k=1}^{9}\Delta_k, && j = 2\kappa+1, \ldots, 3\kappa,\\
&\text{Groupe 4 :} && \beta^{(j)} = -\sqrt{2}\sum_{k=1}^{3}\Delta_k, && j = 3\kappa+1, \ldots, p,
\end{aligned}$$

où $b_j = (-1)^j \frac{1+j \bmod \kappa}{\kappa}$, les fonctions $\Delta_1, \ldots, \Delta_9$ désignent l'ensemble des fonctions définies par :

$$\Delta_s(t) = (1 - 0.2(10t - s)^2)_+,$$

avec $(.)_+$ est la fonction partie positive. Dans ce cadre, seul le troisième groupe a des fonctions coefficients différentes.

(c) Les variables explicatives et la variable à expliquer.
Pour $j = 1, \ldots, p$, $X^{(j)}$ est généré comme

$$X^{(j)}(t) = \sum_{s=1}^{9} a_s \Delta_s(t),$$

où $a_s \sim \mathcal{N}(0, 1), s = 1, \ldots, 9$ et $t \in [0, 1]$.
Ensuite Y est donné par

$$Y = \langle X, \beta \rangle_{\mathcal{H}} + \epsilon,$$

où $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$. Nous choisissons les valeurs de σ_ϵ^2 de telle sorte que le ratio signal sur bruit, $\dfrac{\sigma_\epsilon^2}{var(Y)}$, est à peu près de 10%.

(d) Les deux scénarios de simulation
Deux scénarios sont présentés en fonction de la taille des groupes, κ :

(S1) $\kappa = 3$, $\sigma_\epsilon = 1.6$,

(S2) $\kappa = 20$, $\sigma_\epsilon = 3.6$.

La Figure 4.5 représente les valeurs des $\beta^{(j)}$, $j = 1, \ldots, p$ pour les deux scenarios.

La fonction $\mathbf{X}$ est observée sur 100 points équidistants dans l'intervalle $[0, 1]$. Pour toutes les dimensions de $\mathbf{X}$, $X^{(j)}$ $j = 1, \ldots, p$, nous utilisons comme approximation leur expansion dans une base cubique B-splines de taille $M = 20$. Pour évaluer les performances du modèle, un échantillon d'apprentissage aléatoire de 80% des données est considéré et les 20% restants sont utilisés pour la prédiction. Cette expérience est répétée $I = 100$ fois.

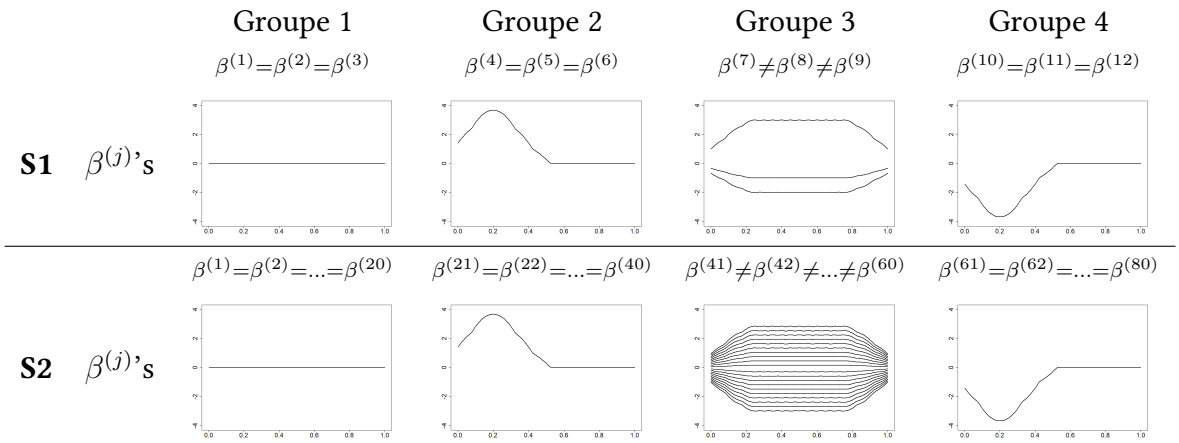

Figure 4.5 – Les valeurs de β suivant les deux scenarios

(e) Les méthodes concurentes
La méthodologie *variable fusion* (FU) est estimée en utilisant la relation 1-NN entre les conditions, tandis que la structure de groupe est utilisée pour le *group fusion lasso* (GFUL). Afin d'évaluer leurs performances, les méthodes FU et GFUL sont comparées à deux méthodes *group lasso* (Godwin, 2013). La première, désignée par GL1 ("Group Lasso 1"), utilise chaque dimension $X^{(j)}$ de $\mathbf{X}$ comme un groupe, comme dans le cadre classique du lasso. La seconde, désignée par GL2 (Group Lasso 2), utilise les mêmes définitions de groupe que dans GFUL (voir l'équation (4.8)). En plus de ces méthodes, nous proposons également le modèle de régression HG (Homogeneous Groups) qui résume toutes les conditions au sein d'un groupe $\mathcal{I}_k$ par leur fonction moyenne,

$$m^{(k)} = \frac{1}{p_k} \sum_{j \in \mathcal{I}_k} X^{(j)},$$

puis estime un modèle linéaire fonctionnel multivarié

$$Y = \sum_{k=1}^{K} \int_0^T m^{(k)}(t)\gamma^{(k)}(t)dt + \epsilon,$$

en utilisant la méthodologie de régression en composantes principales (Aguilera et al. (2006)).

L'idée sous-jacente à cette méthode est d'obtenir le même coefficient de régression pour toutes les conditions au sein d'un groupe : $k = 1, \ldots, K$,

$$\beta^{(j)} = (1/p_k)\gamma^{(k)}, \quad \forall j \in \mathcal{I}_k.$$

La différence avec GFUL est que ce dernier permet à certains groupes d'avoir des fonctions coefficient identiques, alors que HG l'impose à tous les groupes.

À l'exception du modèle HG, qui ne comporte pas de terme de pénalité, les hyperparamètres (α, λ) sont ajustés par validation croisée à 10 folds : λ est choisi dans l'ensemble

$$\lambda \in \left\{0.96^i \lambda_{\max}, i = 0, 1, \ldots, 148\right\} \cup \{0\}$$

et

$$\alpha \in \{0.1, 0.2, \ldots, 1\},$$

avec $\lambda_{\max}$ déterminé de la même manière que dans Lukas and Meier (2020).

(f) La qualité de l'ajustement et autres mesures
Pour chaque méthode, la qualité de l'ajustement est évaluée par l'erreur quadratique moyenne (MSE) calculée sur l'ensemble de test. Leur capacité à retrouver de véritable égalité entre les fonctions coefficient $\beta^{(j)}$ est mesurée par les mesures de "sensibilité" (Sens) et de "spécificité" (Spec). Pour chaque paire $\left(\beta^{(j)}, \beta^{(k)}\right)$, $j, k = 1, \ldots, p$, elles sont définies comme suit,

$$Sens(j,k) = \mathbb{P}\left(\hat{\beta}^{(j)} = \hat{\beta}^{(k)} | \beta^{(j)} = \beta^{(k)}\right),$$

et

$$Spec(j,k) = \mathbb{P}\left(\hat{\beta}^{(j)} \neq \hat{\beta}^{(k)} | \beta^{(j)} \neq \beta^{(k)}\right).$$

Ainsi, $Sens(j,k)$ mesure la capacité de la méthode à obtenir des fonctions coefficient estimées identiques $\hat{\beta}^{(j)} = \hat{\beta}^{(k)}$ lorsque les fonctions théoriques vérifient l'égalité, $\beta^{(j)} = \beta^{(k)}$.

En tant que mesures globales, nous considérons leurs moyennes

$$\text{Sens} = \frac{2}{p(p-1)} \sum_{j=1}^{p} \sum_{k<j} Sens(j,k),$$

$$\text{Spec} = \frac{2}{p(p-1)} \sum_{j=1}^{p} \sum_{k<j} Spec(j,k).$$

4.3.2 Résultats

Scénario 1

Rappelons que dans ce scénario $p = 12$ et $\kappa = 3$. Le résumé des métriques obtenues dans **S1** est présenté dans le tableau 4.1.

Tous les modèles donnent des résultats proches en termes de MSE, sauf le modèle naïf (HG). En effet, ce dernier donne le MSE le plus élevé et l'estimation des fonctions coefficients n'est pas cohérente avec les vraies valeurs de β (voir Figure 4.6). Ainsi, l'hypothèse naïve selon laquelle « les dimensions d'un même groupe partagent la même fonction coefficient de régression » conduit à de mauvais résultats. Le tableau 4.1 montre que les méthodes proposées atteignent les scores de sensibilité et de spécificité les plus élevés. Cela démontre la capacité de ces méthodologies à trouver de véritables égalités entre les coefficients par rapport aux méthodes *group lasso* (GL1 et GL2).

	MSE	Sens	Spec
GL1	6.2(1.59)	0.22(0.13)	1(0.01)
GL2	6.07(1.45)	0.29(0.12)	1(0)
FU	5.97(1.52)	0.82(0.22)	0.99(0.01)
GFUL	5.21(1.81)	0.92(0.22)	1(0)
HG	14.85(3.25)	1(0)	0.95(0)

Table 4.1 – Scénario S1 : moyenne et écart-type (entre parenthèses), des mesures MSE, sensibilité et spécificité obtenues durant les $I = 100$ expériences.

Scénario 2

Le tableau 4.2 présente les résultats dans ce scénario, où $p = 80$ et $\kappa = 20$. Il montre que GFUL est la meilleure méthodologie d'après la métrique MSE. La haute spécificité et la faible sensibilité des méthodes concurrentes indiquent qu'elles ignorent que certains groupes partagent les mêmes fonctions coefficient de régression (voir Figure 4.7). Cela se reflète également dans les critères MSE.

Observons que toutes les autres méthodes, y compris FU, donnent des résultats assez mauvais par rapport à GFUL. Cela peut s'expliquer par le fait que ces méthodes ne sont clairement pas adaptées pour prendre en compte la structure de groupe des conditions.

	MSE	Sens	Spec
GL1	88.74(23.75)	0.2(0.19)	0.85(0.18)
GL2	64.29(17.22)	0.08(0.14)	1(0)
FU	70.58(18.53)	0.07(0.01)	1(0)
GFUL	31.68(16.33)	0.73(0.4)	1(0)
HG	69.37(15.61)	1(0)	0.93(0)

Table 4.2 – Scénario S2 : moyenne et écart-type (entre parenthèses), des mesures MSE, sensibilité et spécificité obtenues durant les $I = 100$ expériences.

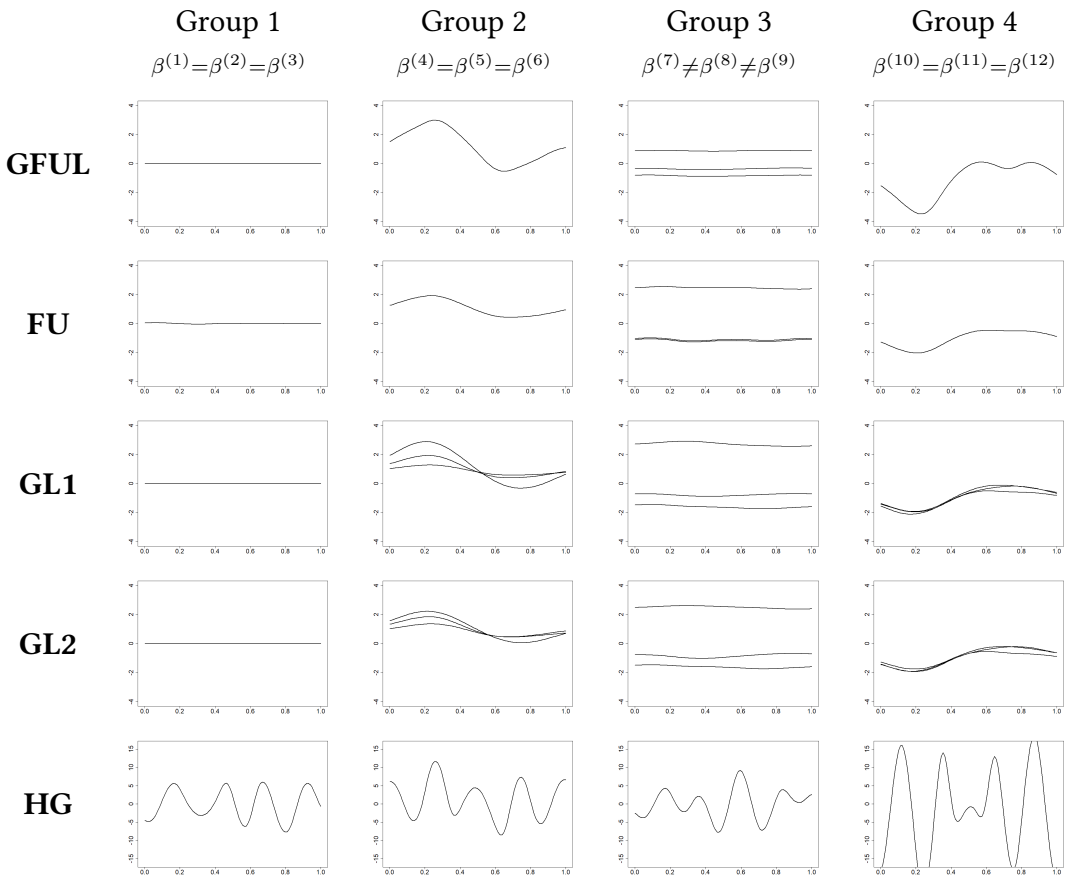

Figure 4.6 – Scénario 1- Les estimations de β par les différentes méthodes (première expérience).

4.4 Application aux données réelles : FingerMovements

Dans cette section, nous appliquons nos méthodes sur l'ensemble de données FingerMovements [1]. Ces données proviennent du domaine de l'interface neuronale directe (ou *brain computer interface*) et sont utilisées pour la classification binaire. Plus précisément, il a été demandé à un sujet de taper des caractères en utilisant uniquement l'index et l'auriculaire de la main droite ($Y = 0$) ou de la main gauche ($Y = 1$). Le défi consiste à déterminer, sur la base de leur enregistrement d'électroencéphalographie (EEG) ($\mathbf{X}$), la main que le sujet a utilisée. Le signal EEG est enregistré pendant une durée de 500 ms par $p = 28$ capteurs situés sur le cuir chevelu. Ainsi, pour chaque sujet, $p = 28$ courbes sont disponibles. Chaque courbe est résumée par 50 points équidistants dans l'intervalle temporel $[0, 500ms]$. La Figure 4.1 (voir la section Introduction) présente un échantillon de courbes enregistrées par 6 capteurs (F1,F2, F3, F4,O1,O2), pour un sujet donné. L'ensemble de données est composé de $N = 416$ sujets et il est divisé en un ensemble d'entraînement de $n = 316$ unités et un ensemble de test de 100 unités.

1. https://www.timeseriesclassification.com/description.php?Dataset=FingerMovements

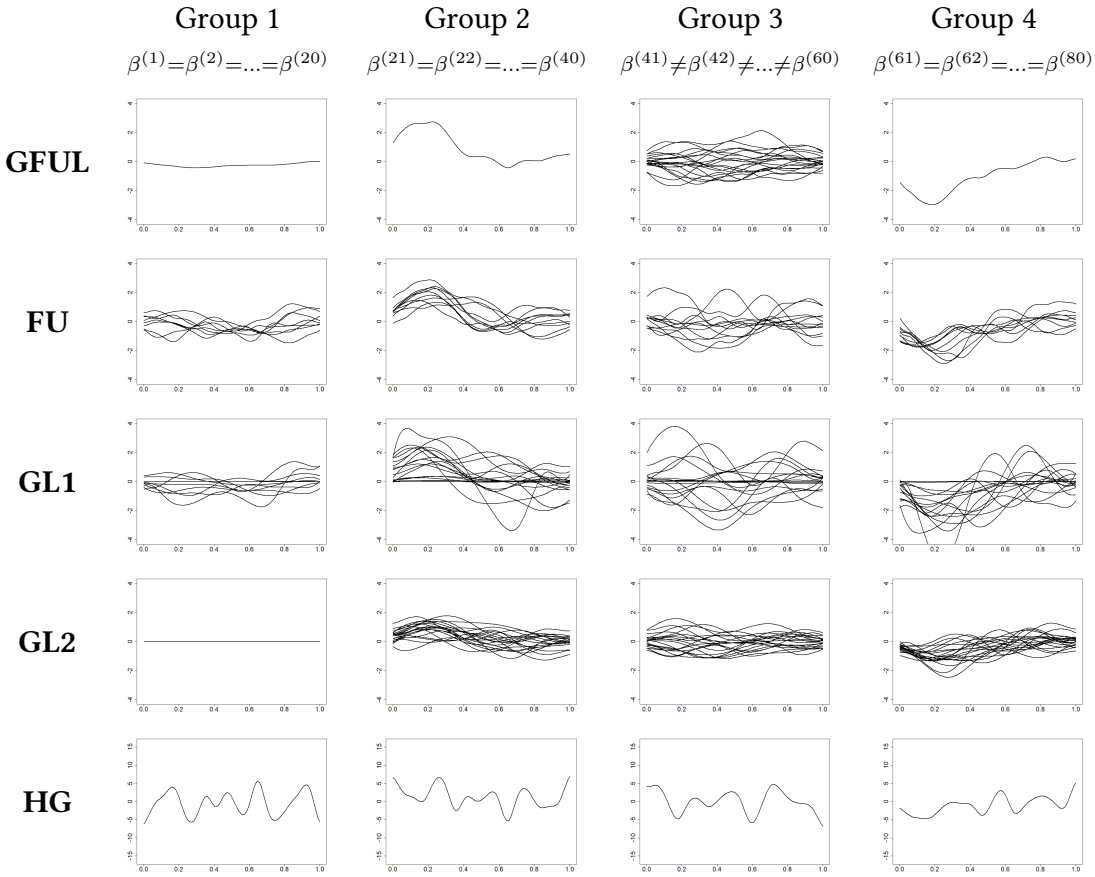

Figure 4.7 – Scénario 2- Les estimations de β par les différentes méthodes (première expérience).

Cet ensemble de données a été utilisé dans Ruiz et al. (2021), où les auteurs ont montré que le modèle Inception Time (IT) (Ismail Fawaz et al., 2020) fournit les meilleures prédictions parmi des modèles de l'état de l'art.

Dans cette section, nous comparons les résultats obtenus par nos méthodologies (FU et GFUL) avec celles des concurrents, à savoir GL1, GL2 et IT.

La méthode FU est basée sur le graphe 1-NN construit à l'aide de la distance euclidienne sur les localisations des capteurs $\mathcal{C}_j \in \mathbb{R}^3$, $j = 1, \ldots, 28$. Pour la méthode GFUL, nous avons utilisé $K = 10$ groupes de conditions obtenus à partir de l'algorithme des K-Means appliqué aux emplacements des capteurs. Les groupes $K = 10$ correspondent à des régions spatiales bien définies (groupe 1 = frontal gauche, groupe 2 = frontal droit, etc). Voir la figure 4.8.

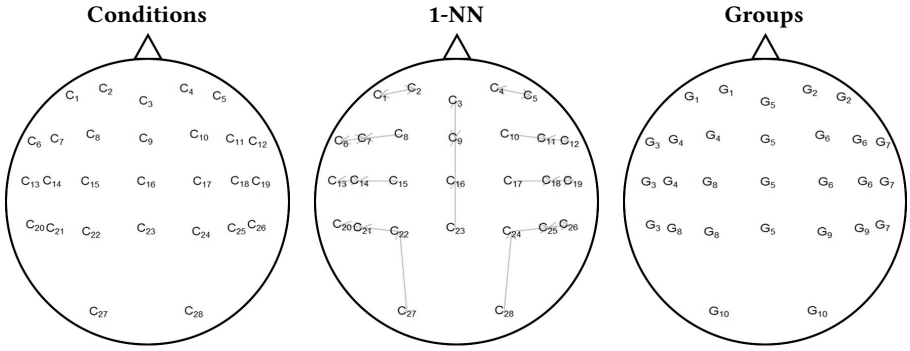

Figure 4.8 – Conditions et groupes pour l'ensemble de données FingerMovements

Deux modèles de *group lasso* sont également estimés : GL1 et GL2. À l'instar de l'étude de simulation, la méthode GL1 utilise chaque dimension comme un groupe tandis que GL2 utilise la même structure de groupe que GFUL.

Pour toutes les dimensions $X^{(j)}$, $j = 1, \ldots, 28$, une base de $M = 30$ B-splines est utilisée pour reconstruire leurs formes fonctionnelles. Les hyperparamètres λ et α sont ajustés par une procédure de validation croisée à 10 folds, sur les grilles suivantes

$$\lambda \in \left\{0.96^i \lambda_{\max}, i = 0, 1, \ldots, 148\right\} \cup \{0\}$$

et

$$\alpha \in \{0, 0.1, 0.2, \ldots, 1\},$$

où $\lambda_{\max}$ est la valeur minimale telle que le terme de pénalité est nul ($\mathcal{P}(\hat{\beta}_{\lambda,\alpha}) = 0$).

4.4.1 Résultats

Le tableau 4.3 montre que les méthodologies proposées donnent souvent de meilleurs résultats par rapport à leurs concurrents (GL2 et IT) en termes de précision (taux de bien classé) sur l'échantillon test. La figure 4.9 montre la structure des fonctions coefficients estimées obtenues avec FU et GFUL. Ainsi, ces résultats fournissent des informations sur l'importance des capteurs et de leur emplacement (également via la structure de groupe) pour la prédiction de la variable réponse. Les graphiques des fonctions coefficients associés sont reléguées en annexe 4.5.

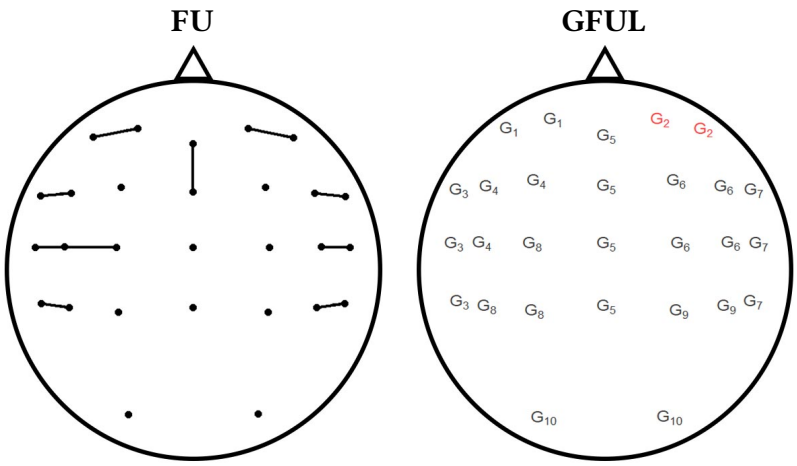

Figure 4.9 – Structures estimées
FU : les points connectés partagent les mêmes coefficients, **GFUL** : les groupes en rouge ont le même coefficient

Méthodes	Précisions
FU	64%
GFUL	68%
IT	56.7%
GL1	65%
GL2	58%

Table 4.3 – Précision obtenue sur le jeu de données de test

4.5 Conclusion et perspectives

Dans ce chapitre, nous avons présenté les méthodes de régression linéaires : *group fusion lasso* (GFUL) et *variable fusion* (FU) pour des variables explicatives fonctionnelles répétées. Ces modèles sont proposés comme un moyen d'intégrer les informations apportées par les conditions $\mathcal{C}_1, \ldots, \mathcal{C}_p$. Par rapport au modèle linéaire classique, ces méthodes permettent de faciliter l'interprétation de la fonction coefficient tout en fournissant de bonnes performances, comme montré par les applications numériques.

Les deux méthodes présentées sont compétitives entre elles et avec les approches existantes.
Le modèle *group fusion lasso* peut être vue comme une généralisation de la méthode *variable fusion* à plusieurs voisins. Cependant, celui-ci teste les relations au sein d'un groupe en une seule fois au lieu de tester de multiples interactions individuelles. C'est une hypothèse forte. Elle suppose que les relations dans un groupe peuvent être soit toutes vraies, soit toutes fausses. L'utilisation de groupes plus petits possiblement superposés pourrait conduire à un modèle alternatif intéressant.

Dans ce cadre, le problème résultant est lié au *group lasso* avec chevauchement de groupes (Yuan et al., 2011), un problème beaucoup plus complexe que les précédents. Une

étude approfondie axée sur la recherche de solution optimale de ce problème doit être menée. Il est aussi possible d'explorer le modèle alternatif *group lasso* proposé dans Jacob et al. (2009). Par contre, il semble que l'utilisation de cette approche conduise à la perte de la diffusion entre les groupes qui se chevauchent. La pénalité n'étant plus définie sur la norme $2, 1$ (voir Jacob et al. (2009) pour plus de détails).

Ici, pour l'ajustement des hyperparamètres, nous avons utilisé des procédures de validation croisée. Ce type de technique ne prend en compte que la performance du modèle. Afin de considérer la complexité des modèles, les alternatives telles que l'utilisation des critères AIC ou BIC peuvent être explorées. Les travaux s'intéressant sur le degré de liberté des modèles lasso tels que Tibshirani and Taylor (2012), Zou et al. (2007) pourraient servir à l'implémentation de ces critères.

Pour le modèle *variable fusion*, nous avons porté notre attention uniquement sur le graphe 1-NN, principalement pour sa praticité et ses propriétés à l'endroit de la redondance des relations. Dans le futur, l'intégration d'autres types de voisinage pourra être explorée. Une étude comparée plus détaillée de ce modèle avec une version *fused lasso* (Tibshirani et al., 2005) pouvant prendre en compte des variables explicatives fonctionnelles devra aussi être menée.

Annexes

.1 Figures supplémentaires (coefficients estimés)

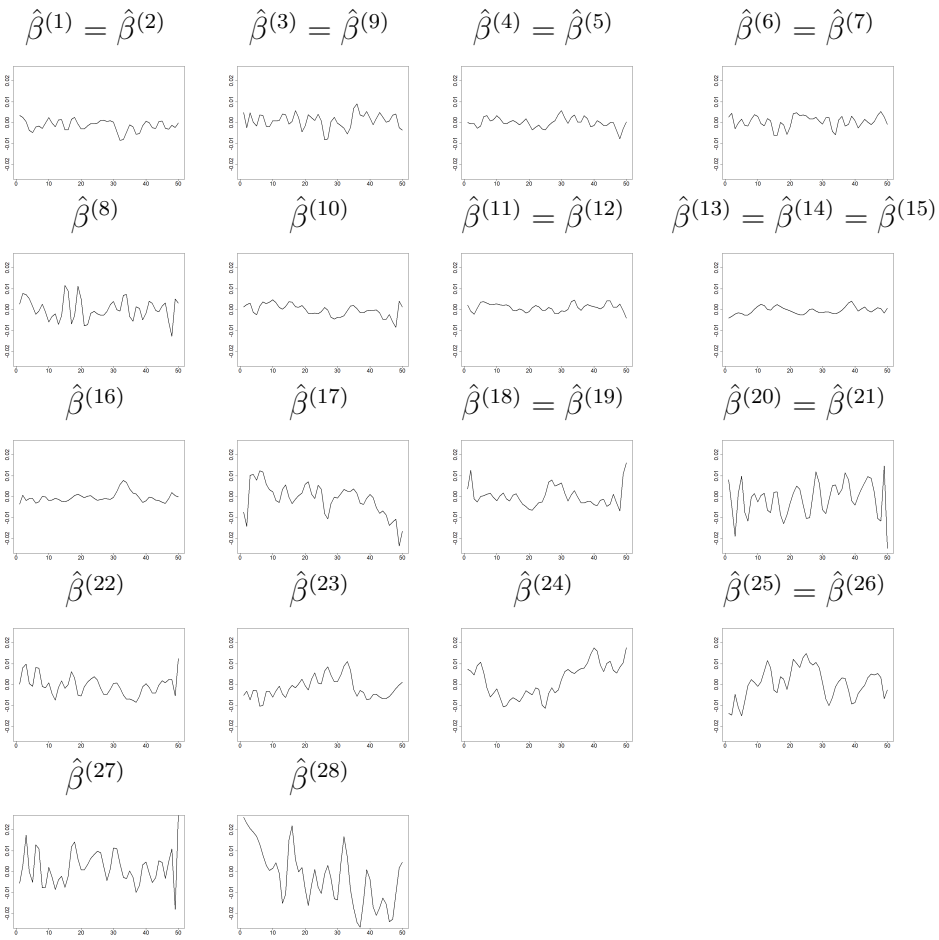

FIGURE 10 – Coefficients estimés par FU

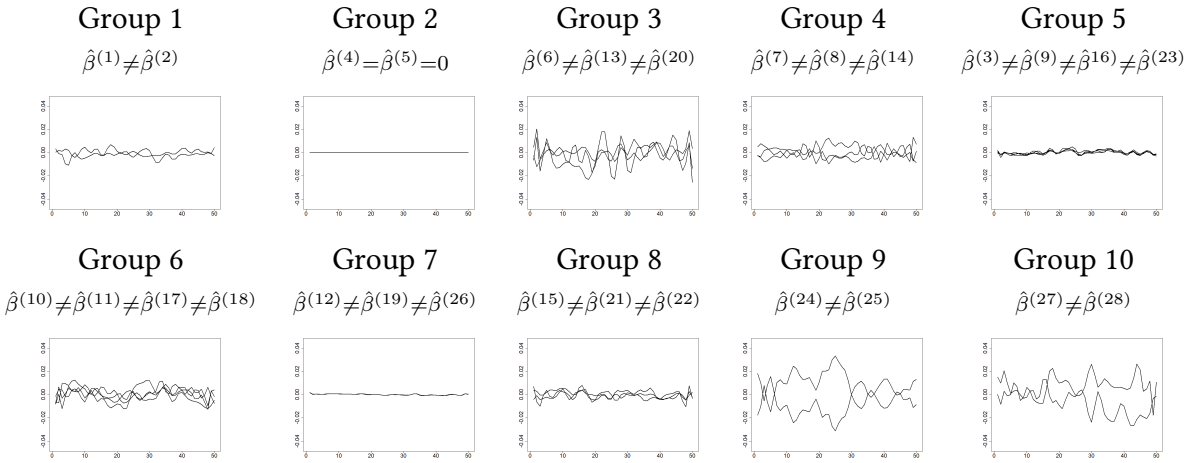

Figure 11 – Coefficients estimés par GFUL

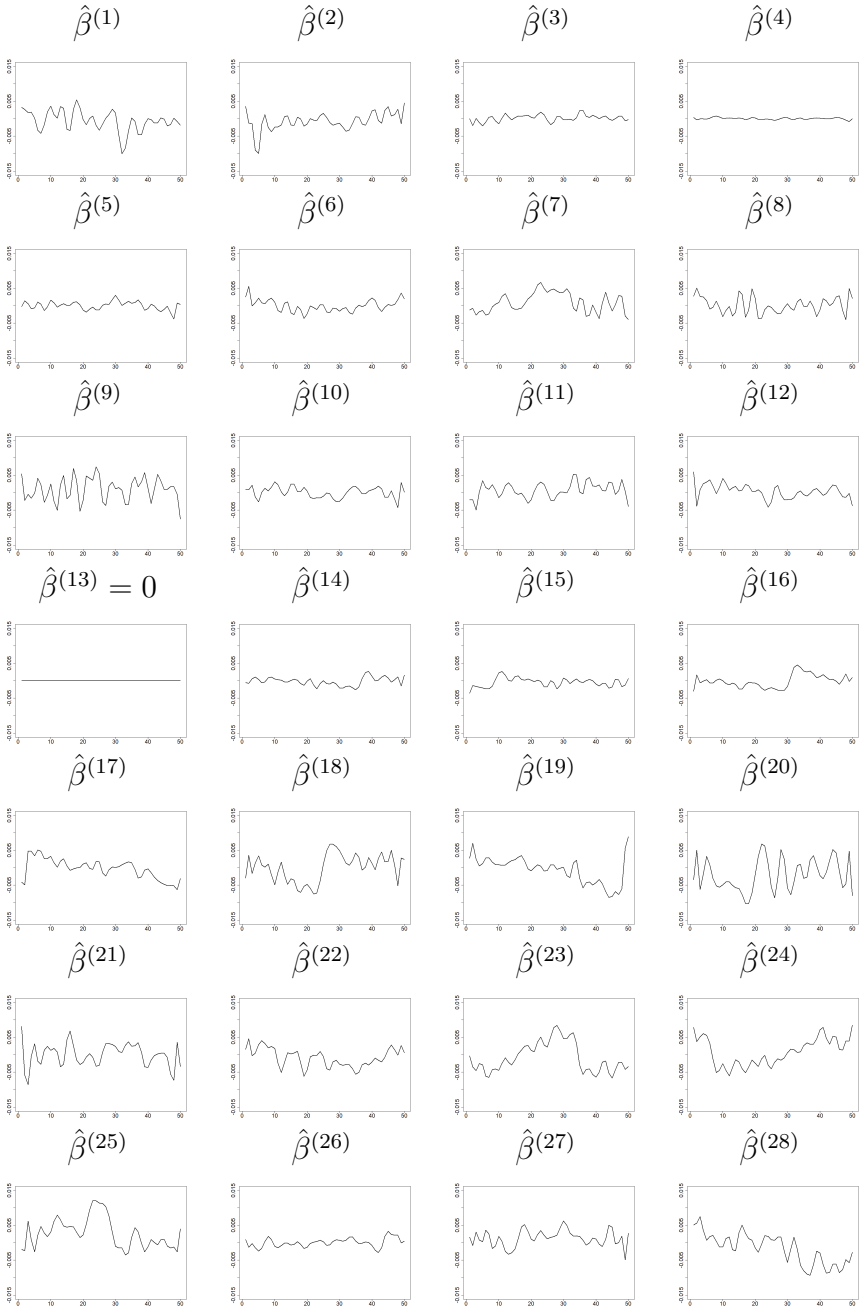

FIGURE 12 – Coefficients estimés par GL1

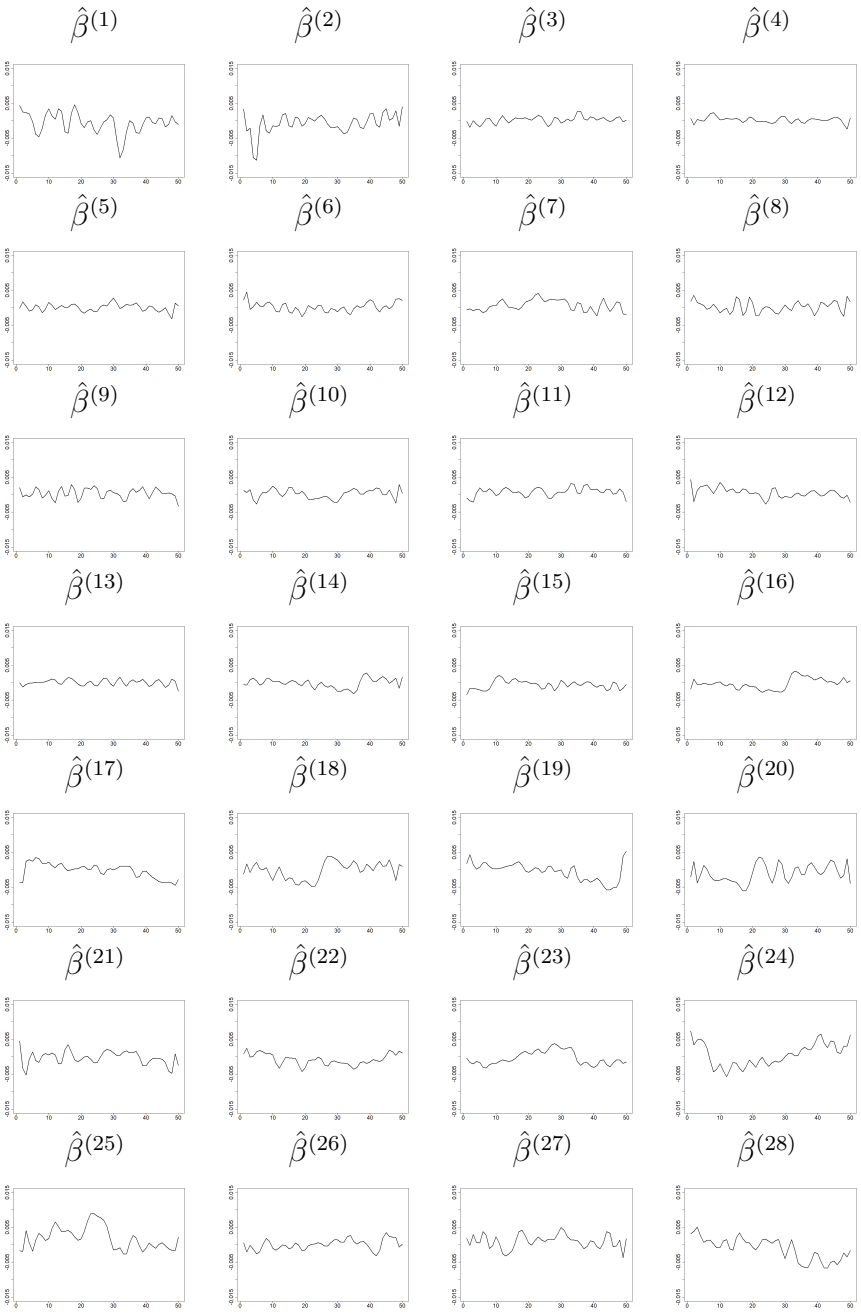

Figure 13 – Coefficients estimés par GL2

.2 Démonstrations

***Démonstration du Lemme** 2.* Soit la fonction de voisinage $v : \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_p\} \to \{1, 2, \ldots, p\}$, telle que définie dans la section 4.2.1 et $\mathcal{V}_0$, $\mathcal{V}_1$ définies comme :

$$\mathcal{V}_0 = \{i \in \{1, 2, \ldots, p\}, v^2(\mathcal{C}_i) = i\}$$

et

$$\mathcal{V}_1 = \{i \in \{1, 2, \ldots, p\}, i > v(\mathcal{C}_i), v^2(\mathcal{C}_i) = i\},$$

avec $v^2(\mathcal{C}_i)$= $v(\mathcal{C}_{v(\mathcal{C}_i)})$, pour $i = 1, \ldots, p$.
Notons que $i \in \mathcal{V}_0$ équivaut à dire que $v(\mathcal{C}_i) \in \mathcal{V}_0$, et que $i \in \mathcal{V}_1$ implique que $v(\mathcal{C}_i) \notin \mathcal{V}_1$. En d'autres termes, $\mathcal{V}_0$ est l'ensemble d'indices correspondant aux conditions pour lesquels une structure 2-cycle est présente. $\mathcal{V}_1$ est un sous-ensemble de $\mathcal{V}_0$ avec card$\mathcal{V}_1 = \frac{1}{2}$card$(\mathcal{V}_0)$. Ainsi, pour $f \in \mathcal{H}$, on a

$$\begin{aligned}\sum_{j=1}^{p} ||(\mathbf{L}f)^{(j)}||_2 &= \sum_{j \in \mathcal{V}_0} ||f^{(j)} - f^{(v(\mathcal{C}_j))}||_2 + \sum_{j \notin \mathcal{V}_0} ||f^{(j)} - f^{(v(\mathcal{C}_j))}||_2 \\ &= \sum_{j \in \mathcal{V}_1} ||2(f^{(j)} - f^{(v(\mathcal{C}_j))})||_2 + \sum_{j \notin \mathcal{V}_0} ||f^{(j)} - f^{(v(\mathcal{C}_j))}||_2.\end{aligned}$$

Alors, il existe une matrice $\mathbf{L}_0 \in \mathbb{R}^{r \times p}$, avec $r = p - \frac{1}{2}$card$(\mathcal{V}_0)$, telle que

$$\sum_{j=1}^{p} ||(\mathbf{L}f)^{(j)}||_2 = \sum_{j=1}^{r} ||(\mathbf{L}_0 f)^{(j)}||_2.$$

Comme v est construite selon les 1-NN, seulement les 2-cycles peuvent être présents (Eppstein et al., 1997)), alors $\mathbf{L}_0$ est de rang maximal. □

Démonstration de la Proposition 3. Un raisonnement similaire à Tibshirani and Taylor (2011) pour le cas des matrices de rang maximal dans le lasso généralisé est utilisé ici. Remarquons d'abords que

$$||\mathbf{D}f||_{L_2,1} = \sum_{j=1}^{r} ||(\mathbf{L}_0 f)^{(j)}||_{L_2} + \sum_{j=r+1}^{p} ||(\mathbf{T}f)^{(j)}||_{L_2}, \quad f \in \mathcal{H}.$$

L'utilisation de la définition de $\mathbf{L}_0$ donne que

$$||\mathbf{D}f||_{L_2,1} = ||\mathbf{L}f||_{L_2,1} + \sum_{j=r+1}^{p} ||(\mathbf{T}f)^{(j)}||_{L_2,1}.$$

Alors, le problème (4.3) peut être reformulé comme

$$\hat{\beta}_\lambda = \arg\min_{f \in \mathcal{H}} \frac{1}{2} \sum_{i=1}^{n} (y_i - \langle x_i, f \rangle_{\mathcal{H}})^2 + \sum_{j=1}^{p} \lambda \mathbb{I}(j \leq r) ||(\mathbf{D}f)^{(j)}||_{L_2} \tag{21}$$

avec $\mathbb{I}(\cdot)$ est la fonction indicatrice. L'inversibilité de $\mathbf{D}$ implique que

$$\hat{\psi}_\lambda = \arg\min_{f\in\mathcal{H}} \frac{1}{2}\sum_{i=1}^{n}\left(y_i - \langle x_i, \mathbf{D}^{-1}f\rangle_{\mathcal{H}}\right)^2 + \sum_{j=1}^{p}\lambda\mathbb{I}(j\leq r)||f^{(j)}||_{L_2}, \tag{22}$$

avec $\hat{\psi}_\lambda = \mathbf{D}\hat{\beta}_\lambda$. L'équation $\langle(\mathbf{D}^{-1})^\top x_i, f\rangle_{\mathcal{H}} = \langle x_i, \mathbf{D}^{-1}f\rangle_{\mathcal{H}}$ permet de conclure la démonstration. □

***Démonstration du lemme** 3.* Afin de simplifier la notation, notons $f_{\mathcal{I}_k}$ la fonction composée uniquement des dimensions de $f \in \mathcal{H}$ qui appartiennent à $\mathcal{I}_k$.

La preuve du lemme repose sur les deux énoncés suivants :

(a) Pour $f \in \mathcal{H}$,

$$||f_{\mathcal{I}_{p_k}} - \bar{f}_{\mathcal{I}_k}\vec{1}_{p_k}||_{L_2,2} = ||\mathrm{R}_k f_{\mathcal{I}_k}||_{L_2,2}.$$

pour tout $k \in \{1,\ldots,K\}$.

(b) Les matrices $\bar{\mathbf{M}}$ et $\mathbf{R}$ vérifient $\mathbf{R}\bar{\mathbf{M}}^\top = \mathbf{0}_{(p-K)\times K}$

Pour le premier point (a), on peut montrer que

$$f_{\mathcal{I}_k} - \bar{f}_{\mathcal{I}_k}\vec{1}_{p_k} = \underbrace{\left[\mathbb{I}_{p_k\times p_k} - \frac{1}{p_k}\mathbf{1}_{p_k\times p_k}\right]}_{\mathrm{P}_k} f_{\mathcal{I}_k}. \tag{23}$$

Le rang de P_k est $p_k - 1$. Soit R_k la matrice R $(p_k-1)\times|p_k|$ obtenue par la décomposition QR de P_k. Comme $||\cdot||_{L_2,2}$ est la norme (fonctionnelle) de Frobenius, on a (a), i.e. $||\mathrm{P}_k f_{\mathcal{I}_k}||_{L_2,2} = ||\mathrm{R}_k f_{\mathcal{I}_k}||_{L_2,2}$.
Pour le point (b), sans perte de généralités, supposons que

$$\mathbf{M} = \begin{pmatrix} \vec{1}_{p_1}^\top & \vec{0}_{p_2}^\top & \ldots & 0 \\ \vec{0}_{p_1}^\top & \vec{1}_{p_2}^\top & \ldots & 0 \\ \ldots & & & \ldots \\ \vec{0}_{p_1}^\top & \vec{0}_{p_2}^\top & \ldots & \vec{1}_{p_K}^\top \end{pmatrix}.$$

Remarquons que $\vec{1}_{p_k}$ appartient au kernel de P_k, i.e $\mathrm{P}_k\vec{1}_{p_k} = \vec{0}_{p_k}$, pour tout $k \in \{1,\ldots,K\}$. À partir de la définition de R_k, on déduit que $\mathrm{P}_k = \mathrm{Q}_k\begin{pmatrix}\mathrm{R}_k \\ \mathbf{0}_{p_k}^\top\end{pmatrix}$, où Q_k est une matrice orthogonale. Ainsi, $\mathrm{P}_k\vec{1}_k = \mathbf{0}_{p_k}$ implique que $\mathrm{R}_k\vec{1}_k = \mathbf{0}_{p_k-1}$. Comme $\mathrm{R}_k\vec{1}_k = \mathbf{0}_{p_k-1}$ pour tout $k \in \{1,\ldots,K\}$, on a

$$\mathbf{R}\bar{\mathbf{M}}^\top = \mathbf{0}_{(p-K)\times K}.$$

En fin, comme conséquence directe de (a), la matrice $\mathbf{G}_\alpha$ satisfait la relation (4.11). Remarquons que $\mathbf{G}_\alpha$ est inversible comme conséquence de (b). Ce qui conclut la démonstration. □

Démonstration de la Proposition 4. La matrice $\mathbf{G}_\alpha$ permet de reformuler le problème 4.10 comme

$$\hat{\beta}_\lambda = \underset{f \in \mathcal{H}}{\arg\min} \frac{1}{2} \sum_{i=1}^{n} \left(y_i, \langle x_i, f \rangle_{\mathcal{H}}\right)^2 + \lambda \sum_{j=1}^{2g} ||(\mathbf{G}_\alpha f)^{(j)}||_{L_2,2}.$$

Comme $\mathbf{G}_\alpha$ est inversible, le résultat est direct. □

Démonstration de la Proposition 5.

1. L'équation (4.14) implique que pour chaque dimension j, $j = 1, \ldots, p$, on a

$$\beta^{(j)}(t) = (b^{(j)})^\top \phi(t),$$

où $b^{(j)} \in \mathbb{R}^M \quad t \in [0, T]$.
Soit F= $\{\langle \phi_i, \phi_j \rangle\}_{i,j}$ et $\mathrm{F} = (\mathrm{F}^{1/2})^\top \mathrm{F}^{1/2}$. Alors, on a que

$$||\beta^{(j)}||_{L_2} = ||(\mathrm{F}^{1/2})^\top b^{(j)}||_2 = ||(b^{(j)})^\top \mathrm{F}^{1/2}||_2.$$

De plus,

$$\mathbf{B}\mathrm{F}^{1/2} = \begin{pmatrix} (b^{(1)})^\top \\ (b^{(2)})^\top \\ \ldots \\ (b^{(p)})^\top \end{pmatrix} \mathrm{F}^{1/2} = \begin{pmatrix} (b^{(1)})^\top \mathrm{F}^{1/2} \\ (b^{(2)})^\top \mathrm{F}^{1/2} \\ \ldots \\ (b^{(p)})^\top \mathrm{F}^{1/2} \end{pmatrix},$$

et $||\beta||_{L_2,1} = ||\mathbf{B}\mathrm{F}^{1/2}||_{2,1}$.

2. Notons que $a_i = \mathrm{vec}(\mathbf{A}_i^\top)$, $b = \mathrm{vec}(\mathbf{B}^\top)$ avec vec(·) est l'opérateur de vectorisation. Cela implique que

$$\begin{aligned} b_0 &= \mathrm{vec}((\mathbf{Z}\mathbf{B})^\top) = \mathrm{vec}(\mathbf{B}^\top \mathbf{Z}^\top) \\ &= (\mathbf{Z} \otimes \mathrm{I}_{M \times M})\mathrm{vec}(\mathbf{B}^\top) = (\mathbf{Z} \otimes \mathrm{I}_{M \times M})b, \end{aligned}$$

où $\otimes$ est le produit de Kronecker.

□

Chapitre 5

Conclusion et perspectives

5.1 Conclusion

Les avancées technologiques sur le stockage et les appareils de mesures ont favorisé, de plus en plus, la collection de données fonctionnelles multivariées. Le sujet de ce document : l'apprentissage supervisé, est une problématique centrale relative à ces données. Dans ce cadre, nous avons fait le choix de nous concentrer sur le développement de modèles facilement interprétables. Cela, car l'interprétabilité paraît de plus en plus essentielle comme certaines de ces données proviennent de domaines sensibles (médecine, sécurité, etc.). Ainsi, ce document introduit plusieurs modèles linéaires et un modèle semi-linéaire pour la prédiction d'une variable scalaire à l'aide de variables fonctionnelles multivariées. Le choix de ce type de modèles nous a semblé particulièrement attractif car (en plus de leur potentiel interprétatif) les modèles sont relativement rapides à estimer.

Les données fonctionnelles multivariées peuvent être de plusieurs types, suivant la variable aléatoire dont elles sont les observations. Dans les chapitres 2 et 3, nous nous sommes intéressés au cadre le plus général : les variables fonctionnelles multivariées potentiellement définies sur différents domaines. On peut imaginer, par exemple, que la variable explicative est un vecteur composé de séries temporelles et/ou d'images. L'approche PLS (MFPLS), introduite dans le chapitre 2, permet la classification binaire et la régression sur ce type de vecteur de variables. Les applications numériques sur des données simulées et réelles ont démontré la compétitivité de cette méthode par rapport à des approches existantes. Contrairement à certaines d'entre elles (LSTM, RF, etc.), la méthode MFPLS permet une relative transparence des modèles. En effet, comme elle est basée sur un modèle de régression linéaire fonctionnel, pour interpréter les résultats, il suffit d'étudier la fonction coefficient qu'elle estime.

L'approche MFPLS étant dépendante de l'hypothèse linéaire entre les variables explicatives et la variable d'intérêt, le chapitre 3 propose un modèle plus flexible : TMFPLS. Celui-ci est un arbre de décision qui se base sur la discrimination binaire de l'approche PLS pour scinder les nœuds. Il permet une classification à plus de deux classes avec des vecteurs de variables explicatives fonctionnelles de types différents (images, séries temporelles, etc.). La

comparaison des approches TMFPLS et MFPLS sur des données simulées a montré la grande flexibilité de l'arbre. Il offre de meilleures performances dans un contexte de modèle génératif des données plus complexe qu'une relation linéaire (la classification visuelle). De plus, comme nos modèles (MFPLS et TMFPLS) peuvent inclure des données de types différents, afin d'accroître les performances de nos modèles, on peut intégrer le plan fréquentiel dans la discrimination des classes à l'aide du spectrogramme. Cela est illustré dans le chapitre 3.

Le cadre d'une variable explicative fonctionnelle multivariée définie sur plusieurs domaines est pratique dans la mesure où les modèles, proposés dans ce contexte, peuvent être utilisés dans une majorité des cas : variable univariée, variable multivariée (unique ou différents domaines). Cependant, plus les dimensions de la variable explicative augmentent, plus il devient difficile de donner des interprétations claires des modèles proposés. Alors, le chapitre 3 a traité du cas de variables fonctionnelles répétées explicatives. Celui-ci, souvent simplement considéré comme fonctionnel multivarié, désigne le cas où toutes les dimensions de la variable explicative représentent une unique variable fonctionnelle, mais observée suivant différentes *conditions*. Nous supposons que ces dernières appartiennent à un espace métrique, c'est-à-dire qu'il existe une notion de proximité entre elles. On montre que l'utilisation de ces conditions permet de diminuer la complexité liée à l'interprétation, grâce notamment aux modèles *variable fusion* et *group fusion lasso*, tout en fournissant des résultats compétitifs.

5.2 Perspectives

Comme suite de ce travail, les perspectives pourront s'articuler autour de l'extension des cadres d'application des modèles proposés.

La régression pour TMFPLS Comme pour la classification, la régression à l'aide de l'arbre TMFPLS servirait à obtenir un modèle plus flexible que MFPLS. Pour ce faire, on pourrait se concentrer sur la recherche de la meilleure procédure pour scinder les nœuds, ce qui n'est pas direct pour la régression. En effet, supposons que l'on a à notre disposition le couple de variables (X, Y), où X désigne la variable fonctionnelle multivariée explicative et Y la variable à prédire. Pour simplifier, on dira que X n'as pas de structure de groupe particulière. De manière formelle, scinder le nœud racine de l'arbre (binaire) peut se ramener à trouver la variable latente Z à valeurs dans $\{0, 1\}$ telle que les modèles de régression au sein des nœuds soient les plus performants possibles. Donc, il nous faut estimer la loi du tuple (X, Y, Z), ce qui est possible, par exemple, en utilisant la régression par groupes (Preda and Saporta, 2005). Cependant, l'utilisation de cette méthode pose un problème pour la prédiction (Saporta, 2008). En effet, supposons $x_1 \in \mathcal{H}$ une réalisation d'une variable fonctionnelle de même loi que X, prédire y_1 reviendrait aussi à estimer la probabilité $\mathbb{P}(Z|X = x_1, Y)$, alors que y_1 n'est pas observé et qu'on cherche précisément à l'estimer. Autrement dit, cela revient à prédire les variables Z, Y à l'aide uniquement de X, alors que Z dépend du couple (X, Y).

Une autre possibilité est de ramener la régression à un problème de classification (Torgo and Gama (1996)). Dans ce cas, la principale difficulté réside dans la recherche d'une dis-

crétisation optimale de Y.

La pénalité fusion Pour le modèle *variable fusion*, nous avons fait le choix de prendre la fonction 1-NN pour modéliser les proximités entre conditions. Une conséquence de cette fonction est qu'elle considère que tous les voisins se valent. Autrement dit, elle ne prend pas en compte la distance entre deux voisins. Pour illustrer ce point, considérons la Figure 5.1, qui représente 5 conditions dans $\mathbb{R}$.

Figure 5.1 – Exemples de conditions : $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_5$

La matrice d'adjacence est alors donnée par :

$$\mathbf{W} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}.$$

Elle montre que la condition 2 ($\mathcal{C}_2$) est autant la voisine de $\mathcal{C}_1$ que de $\mathcal{C}_3$. Alors que la Figure 5.1 montre clairement que $\mathcal{C}_2$ est plus proche de $\mathcal{C}_1$ que $\mathcal{C}_3$ ne l'est. Sachant que $\mathbf{W}$ est utilisée pour définir la pénalité fusion, il serait peut-être plus pertinent d'ajouter une mesure afin de tenir compte de la proximité entre deux conditions voisines. Pour cela, on pourrait considérer, par exemple, les distances entre voisins w_j

$$w_j = d(\mathcal{C}_j, \mathcal{C}_{v(\mathcal{C}_j)})$$

pour $j = 1, \ldots, 5$, où d est la mesure de distance entre les conditions et v la fonction 1-NN, il est possible de se référer au chapitre 4 qui en présente les définitions. Une version standardisée de ces distances s'obtient par :

$$w_j^* = \frac{w - w_j}{w}$$

où $w = \sum_j w_j$, avec $w \neq 0$. Ainsi, par exemple, pour $w_1, \ldots, w_4$ fixées, si w_5 est *grand*, alors w_5^* tend vers 0, de même que si w_5 est *petit*, alors w_5^* tend vers 1. L'utilisation de $w_1^*, \ldots, w_5^*$ pourrait permettre de donner plus de poids aux voisins proches par rapport aux autres. Par exemple, on pourrait remplacer la matrice d'adjacence par la matrice $\bar{\mathbf{W}}$, définie de la façon suivante :

$$\bar{\mathbf{W}} = \begin{pmatrix} w_1^* & \ldots & w_1^* \\ \vdots & \ldots & \vdots \\ w_5^* & \ldots & w_5^* \end{pmatrix} \circ \mathbf{W}$$

où $\circ$ est le produit matriciel d'Hadamard. D'une certaine manière, cela revient à faire de l'*adaptative* variable fusion. Le travail Zou (2006) pourra servir de référence pour le développement de cet axe de recherche.

Pour le *group fusion lasso* (GFUL), nous nous sommes limités au cas de groupes de conditions disjoints. Ce cadre est restrictif au regard d'un certain nombre d'applications. Par exemple, les électrodes dans l'EEG peuvent appartenir simultanément à plusieurs zones du cerveau (pariétales, occipitales, etc.). De plus, la prise en compte de groupes potentiellement disjoints permettrait, à l'aide de la diffusion, de découvrir des structures complexes de distributions des coefficients (Yuan et al., 2011).

Variable fonctionnelle catégorielle : A moyen terme, il est aussi possible d'envisager les extensions des méthodes proposées pour des variables fonctionnelles catégorielles (Deville and Saporta (1979), Preda et al. (2021)). Les auteurs dans Preda (2015) proposent une approche pour la régression avec ce type de variable, mais celle-ci se limite au cadre univarié. Il serait intéressant de considérer un cadre plus général, comme des variables multivariées fonctionnelles mixtes, c'est-à-dire de type continues et catégorielles.

En conclusion, les travaux futurs devront se concentrer sur les points mentionnés ci-dessus, tout en tâchant de fournir des études théoriques poussées des méthodes.

Bibliographie

Abd El-Samie, F. E., Alotaiby, T. N., Khalid, M. I., Alshebeili, S. A., and Aldosari, S. A. (2018). A review of eeg and meg epileptic spike detection algorithms. *IEEE Access*, 6 :60673–60688.

Aguilera, A. M., Escabias, M., Preda, C., and Saporta, G. (2010). Using basis expansions for estimating functional pls regression : applications with chemometric data. *Chemometrics and Intelligent Laboratory Systems*, 104(2) :289–305.

Aguilera, A. M., Escabias, M., and Valderrama, M. J. (2006). Using principal components for estimating logistic regression with high-dimensional multicollinear data. *Computational Statistics & Data Analysis*, 50(8) :1905–1924.

Bagnall, A., Flynn, M., Large, J., Lines, J., and Middlehurst, M. (2020). A tale of two toolkits, report the third : on the usage and performance of hive-cote v1. 0. *arXiv preprint arXiv :2004.06069*.

Bastien, P., Vinzi, V. E., and Tenenhaus, M. (2005). Pls generalised linear regression. *Computational Statistics & data analysis*, 48(1) :17–46.

Belli, E. and Vantini, S. (2022). Measure inducing classification and regression trees for functional data. *Statistical Analysis and Data Mining : The ASA Data Science Journal*, 15(5) :553–569.

Besse, P. and Ramsay, J. O. (1986). Principal components analysis of sampled functions. *Psychometrika*, 51(2) :285–311.

Beyaztas, U. and Shang, H. L. (2022). A robust functional partial least squares for scalar-on-multiple-function regression. *Journal of Chemometrics*, page e3394.

Biau, G., Bunea, F., and Wegkamp, M. H. (2005). Functional classification in hilbert spaces. *IEEE Transactions on Information Theory*, 51(6) :2163–2172.

Blanquero, R., Carrizosa, E., Jiménez-Cordero, A., and Martín-Barragán, B. (2019). Variable selection in classification for multivariate functional data. *Information Sciences*, 481 :445–462.

Blanquero, R., Carrizosa, E., Molero-Río, C., and Morales, D. R. (2023). On optimal regression trees to detect critical intervals for multivariate functional data. *Computers & Operations Research*, page 106152.

Bleakley, K. and Vert, J.-P. (2011). The group fused lasso for multiple change-point detection. *arXiv preprint arXiv :1106.4199*.

Bosq, D. (2000). *Linear processes in function spaces : theory and applications*, volume 149. Springer Science & Business Media.

Breiman, L. (1984). *Classification and regression trees*. Routledge.

Cardot, H., Ferraty, F., and Sarda, P. (1999). Functional linear model. *Statistics & Probability Letters*, 45(1) :11–22.

Carnegie (0). Carnegie Mellon University- cmu graphics lab - motion capture library. http://mocap.cs.cmu.edu/. Accessed : 2022-05.

Carrizosa Priego, E. J., Molero Río, C., and Romero Morales, M. D. (2021). Mathematical optimization in classifcation and regression trees. *TOP, 29, 6-33.*

Chen, K. and Müller, H.-G. (2012). Modeling repeated functional observations. *Journal of the American Statistical Association*, 107(500) :1599–1609.

Dauxois, J., Pousse, A., and Romain, Y. (1982). Asymptotic theory for the principal component analysis of a vector random function : some applications to statistical inference. *Journal of multivariate analysis*, 12(1) :136–154.

De Boor, C. (1972). On calculating with b-splines. *Journal of Approximation theory*, 6(1) :50–62.

Delaigle, A. and Hall, P. (2012a). Achieving near perfect classification for functional data. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, 74(2) :267–286.

Delaigle, A. and Hall, P. (2012b). Methodology and theory for partial least squares applied to functional data. *The Annals of Statistics*, 40(1) :322–352.

Dembowska, S., Liu, H., Houwing-Duistermaat, J., and Frangi, A. (2021). Multivariate functional partial least squares for classification using longitudinal data. *Multivariate functional partial least squares for classification using longitudinal data*, pages 75–88.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society : series B (methodological)*, 39(1) :1–22.

Deville, J.-C. and Saporta, G. (1979). L'analyse harmonique qualitative. In *Analyse des données et Informatique*, pages 375–389. North-Holland.

El Haouij, N., Poggi, J.-M., Ghozi, R., Sevestre-Ghalila, S., and Jaïdane, M. (2019). Random forest-based approach for physiological functional variable selection for driver's stress level classification. *Statistical Methods & Applications*, 28 :157–185.

Eppstein, D., Paterson, M. S., and Yao, F. F. (1997). On nearest-neighbor graphs. *Discrete & Computational Geometry*, 17(3) :263–282.

Escabias, M., Aguilera, A., and Valderrama, M. (2004). Principal component estimation of functional logistic regression : discussion of two different approaches. *Journal of Nonparametric Statistics*, 16(3-4) :365–384.

Escabias, M., Aguilera, A., and Valderrama, M. (2005). Modeling environmental data by functional principal component logistic regression. *Environmetrics : The official journal of the International Environmetrics Society*, 16(1) :95–107.

Escabias, M., Aguilera, A. M., and Valderrama, M. J. (2007). Functional pls logit regression model. *Computational Statistics & Data Analysis*, 51(10) :4891–4902.

Febrero-Bande, M., Galeano, P., and González-Manteiga, W. (2017). Functional principal component regression and functional partial least-squares regression : An overview and a comparative study. *International Statistical Review*, 85(1) :61–83.

Ferraty, F. and Vieu, P. (2006). *Nonparametric functional data analysis : theory and practice*, volume 76. Springer.

Fukushima, K. (1988). A neural network for visual pattern recognition. *Computer*, 21(3) :65–75.

Galeano, P., Joseph, E., and Lillo, R. E. (2015). The mahalanobis distance for functional data with applications to classification. *Technometrics*, 57(2) :281–291.

Gardner-Lubbe, S. (2021). Linear discriminant analysis for multiple functional data analysis. *Journal of Applied Statistics*, 48(11) :1917–1933.

Godwin, J. (2013). Group lasso for functional logistic regression. Master's thesis.

Goldsmith, J., Bobb, J., Crainiceanu, C. M., Caffo, B., and Reich, D. (2011). Penalized functional regression. *Journal of computational and graphical statistics*, 20(4) :830–851.

Golovkine, S., Klutchnikoff, N., and Patilea, V. (2022). Clustering multivariate functional data using unsupervised binary trees. *Computational Statistics & Data Analysis*, 168 :107376.

Górecki, T., Krzyśko, M., and Wołyński, W. (2015). Classification problems based on regression models for multi-dimensional functional data. *Statistics in Transition new series*, 16(1).

Grenander, U. (1950). Stochastic processes and statistical inference. *Arkiv för matematik*, 1(3) :195–277.

Greven, S., Crainiceanu, C., Caffo, B., and Reich, D. (2011). Longitudinal functional principal component analysis. In *Recent Advances in Functional Data Analysis and Related Topics*, pages 149–154. Springer.

Guan, T., Lin, Z., Groves, K., and Cao, J. (2022). Sparse functional partial least squares regression with a locally sparse slope function. *Statistics and Computing*, 32(2) :1–11.

Hall, P. and Horowitz, J. L. (2007). Methodology and convergence rates for functional linear regression.

Hall, P., Poskitt, D. S., and Presnell, B. (2001). A functional data—analytic approach to signal discrimination. *Technometrics*, 43(1) :1–9.

Happ, C. (2017). Object-oriented software for functional data. *arXiv preprint arXiv :1707.02129*.

Happ, C. and Greven, S. (2018). Multivariate functional principal component analysis for data observed on different (dimensional) domains. *Journal of the American Statistical Association*, 113(522) :649–659.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8) :1735–1780.

Horváth, L. and Kokoszka, P. (2012). *Inference for functional data with applications*, volume 200. Springer Science & Business Media.

Hu, X. and Yao, F. (2020). Sparse functional principal component analysis in high dimensions. *arXiv preprint arXiv :2011.00959*.

Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., and Muller, P.-A. (2019). Deep learning for time series classification : a review. *Data mining and knowledge discovery*, 33(4) :917–963.

Ismail Fawaz, H., Lucas, B., Forestier, G., Pelletier, C., Schmidt, D. F., Weber, J., Webb, G. I., Idoumghar, L., Muller, P.-A., and Petitjean, F. (2020). Inceptiontime : Finding alexnet for time series classification. *Data Mining and Knowledge Discovery*, 34(6) :1936–1962.

Jacob, L., Obozinski, G., and Vert, J.-P. (2009). Group lasso with overlap and graph lasso. In *Proceedings of the 26th annual international conference on machine learning*, pages 433–440.

Jacques, J. and Preda, C. (2014). Model-based clustering for multivariate functional data. *Computational Statistics & Data Analysis*, 71 :92–106.

James, G. M. and Hastie, T. J. (2001). Functional linear discriminant analysis for irregularly sampled curves. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, 63(3) :533–550.

James, G. M., Wang, J., and Zhu, J. (2009). Functional linear regression that's interpretable.

Javed, R., Rahim, M. S. M., Saba, T., and Rehman, A. (2020). A comparative study of features selection for skin lesion detection from dermoscopic images. *Network Modeling Analysis in Health Informatics and Bioinformatics*, 9(1) :1–13.

Jong, S. D. (1993). Pls fits closer than pcr. *Journal of chemometrics*, 7(6) :551–557.

Kara, L.-Z., Laksaci, A., Rachdi, M., and Vieu, P. (2017). Data-driven knn estimation in nonparametric functional data analysis. *Journal of Multivariate Analysis*, 153 :176–188.

Karhunen, K. (1946). Zur spektraltheorie stochastischer prozesse. *Ann. Acad. Sci. Fennicae, AI*, 34.

Karim, F., Majumdar, S., Darabi, H., and Chen, S. (2017). Lstm fully convolutional networks for time series classification. *IEEE access*, 6 :1662–1669.

Karim, F., Majumdar, S., Darabi, H., and Harford, S. (2019). Multivariate lstm-fcns for time series classification. *Neural Networks*, 116 :237–245.

Kokoszka, P. and Reimherr, M. (2017). *Introduction to functional data analysis*. CRC press.

Koner, S. and Staicu, A.-M. (2023). Second-generation functional data. *Annual Review of Statistics and Its Application*, 10 :547–572.

Land, S. R. and Friedman, J. H. (1997). Variable fusion : A new adaptive signal regression method.

Lee, E. R. and Park, B. U. (2012). Sparse estimation in functional linear regression. *Journal of Multivariate Analysis*, 105(1) :1–17.

Lichman, M. (2013). Uci machine learning repository. http://archive.ics.uci.edu/ml//.

Locantore, N., Marron, J., Simpson, D., Tripoli, N., Zhang, J., Cohen, K., Boente, G., Fraiman, R., Brumback, B., Croux, C., et al. (1999). Robust principal component analysis for functional data. *Test*, 8 :1–73.

Loève, M. (1946). Fonctions aléatoires à décomposition orthogonale exponentielle. *La Revue Scientifique*, 84 :159–162.

López-Pintado, S. and Romo, J. (2006). Depth-based classification for functional data. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 72 :103.

Lukas and Meier (2020). Package 'grplasso'.

Maturo, F. and Verde, R. (2022). Supervised classification of curves via a combined use of functional data analysis and tree-based methods. *Computational Statistics*, pages 1–41.

Meier, L., Van De Geer, S., and Bühlmann, P. (2008). The group lasso for logistic regression. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, 70(1) :53–71.

Moindjié, I.-A., Dabo-Niang, S., and Preda, C. (2022). Classification of multivariate functional data on different domains with partial least squares approaches. *arXiv preprint arXiv :2212.09145*.

Möller, A. and Gertheiss, J. (2018). A classification tree for functional data. In *International workshop on statistical modeling*.

Morris, J. S. (2015). Functional regression. *Annual Review of Statistics and Its Application*, 2 :321–359.

Oleszewski, R. (2012). http://www.cs.cmu.edu/~bobski//.

Pei, W., Dibeklioğlu, H., Tax, D. M., and van der Maaten, L. (2017). Multivariate time-series classification using the hidden-unit logistic model. *IEEE transactions on neural networks and learning systems*, 29(4) :920–931.

Petch, J., Di, S., and Nelson, W. (2022). Opening the black box : the promise and limitations of explainable machine learning in cardiology. *Canadian Journal of Cardiology*, 38(2) :204–213.

Poterie, A., Dupuy, J.-F., Monbet, V., and Rouviere, L. (2019). Classification tree algorithm for grouped variables. *Computational Statistics*, 34 :1613–1648.

Preda, C. (2015). Regression with categorical functional data. In *8th International Conference of the ERCIM WG on Computational and Methodological Statistics*.

Preda, C., Grimonprez, Q., and Vandewalle, V. (2021). Categorical functional data analysis. the cfda r package. *Mathematics*, 9(23) :3074.

Preda, C. and Saporta, G. (2002). Régression pls sur un processus stochastique. *Revue de statistique appliquée*, 50(2) :27–45.

Preda, C. and Saporta, G. (2005). Clusterwise pls regression on a stochastic process. *Computational Statistics & Data Analysis*, 49(1) :99–108.

Preda, C., Saporta, G., and Lévéder, C. (2007). Pls classification of functional data. *Computational Statistics*, 22(2) :223–235.

Ramsay, J. O. (1982). When the data are functions. *Psychometrika*, 47 :379–396.

Ramsay, J. O. and Dalzell, C. (1991). Some tools for functional data analysis. *Journal of the Royal Statistical Society : Series B (Methodological)*, 53(3) :539–561.

Ramsey, J. O. and Silverman, B. W. (2005). *Functional Data Analysis*. Springer-Verlag, 2 edition.

Rao, C. R. (1958). Some statistical methods for comparison of growth curves. *Biometrics*, 14(1) :1–17.

Reiss, P. T., Goldsmith, J., Shang, H. L., and Ogden, R. T. (2017). Methods for scalar-on-function regression. *International Statistical Review*, 85(2) :228–249.

Reiss, P. T. and Ogden, R. T. (2007). Functional principal component regression and functional partial least squares. *Journal of the American Statistical Association*, 102(479) :984–996.

Ribeiro Jr, P. J., Diggle, P. J., et al. (2001). geor : a package for geostatistical analysis. *R news*, 1(2) :14–18.

Ruiz, A. P., Flynn, M., Large, J., Middlehurst, M., and Bagnall, A. (2021). The great multivariate time series classification bake off : a review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery*, 35(2) :401–449.

Saikhu, A., Arifin, A. Z., and Fatichah, C. (2019). Correlation and symmetrical uncertainty-based feature selection for multivariate time series classification. *International Journal of Intelligent Engineering and System*, 12(3) :129–137.

Saporta, G. (1981). Méthodes exploratoires d'analyse de données temporelles. *Cahiers du Bureau universitaire de recherche opérationnelle Série Recherche*, 37 :7–194.

Saporta, G. (2008). Models for understanding versus models for prediction. In *COMPSTAT 2008 : Proceedings in computational statistics*, pages 315–322. Springer.

Schäfer, P. and Leser, U. (2017). Multivariate time series classification with weasel+ muse. *arXiv preprint arXiv :1711.11343*.

Shang, H. L. (2014). A survey of functional principal component analysis. *AStA Advances in Statistical Analysis*, 98 :121–142.

Sørensen, H., Goldsmith, J., and Sangalli, L. M. (2013). An introduction with medical applications to functional data analysis. *Statistics in medicine*, 32(30) :5222–5240.

Sübakan, Y. C., Kurt, B., Cemgil, A. T., and Sankur, B. (2014). Probabilistic sequence clustering with spectral learning. *Digital Signal Processing*, 29 :1–19.

Tenenhaus, M., Gauchi, J.-P., and Ménardo, C. (1995). Régression pls et applications. *Revue de statistique appliquée*, 43(1) :7–63.

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., and Knight, K. (2005). Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, 67(1) :91–108.

Tibshirani, R. J. and Taylor, J. (2011). The solution path of the generalized lasso. *The annals of statistics*, 39(3) :1335–1371.

Tibshirani, R. J. and Taylor, J. (2012). Degrees of freedom in lasso problems.

Torgo, L. and Gama, J. (1996). Regression by classification. In *Advances in Artificial Intelligence : 13th Brazilian Symposium on Artificial Intelligence, SBIA'96 Curitiba, Brazil, October 23–25, 1996 Proceedings 13*, pages 51–60. Springer.

Tuncel, K. S. and Baydogan, M. G. (2018). Autoregressive forests for multivariate time series modeling. *Pattern recognition*, 73 :202–215.

Wang, J.-L., Chiou, J.-M., and Müller, H.-G. (2016). Functional data analysis. *Annual Review of Statistics and its application*, 3 :257–295.

Yao, F., Müller, H.-G., and Wang, J.-L. (2005). Functional data analysis for sparse longitudinal data. *Journal of the American statistical association*, 100(470) :577–590.

Yi, Y., Billor, N., Liang, M., Cao, X., Ekstrom, A., and Zheng, J. (2022). Classification of eeg signals : an interpretable approach using functional data analysis. *Journal of Neuroscience Methods*, 376 :109609.

Yuan, L., Liu, J., and Ye, J. (2011). Efficient methods for overlapping group lasso. *Advances in neural information processing systems*, 24.

Yuan, M. and Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, 68(1) :49–67.

Zhang, X. and Wang, J.-L. (2016). From sparse to dense functional data and beyond.

Zhou, L. and Pan, H. (2014). Principal component analysis of two-dimensional functional data. *Journal of Computational and Graphical Statistics*, 23(3) :779–801.

Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American statistical association*, 101(476) :1418–1429.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B : Statistical Methodology*, 67(2) :301–320.

Zou, H., Hastie, T., and Tibshirani, R. (2007). On the "degrees of freedom" of the lasso.